



Refine Query:

semiconductor and pad and inverter - *pm* -

Search

Help

Collections: Journals Conferences Standards

Your search matched **1** of **506922** documents.

1 are presented on this page, sorted by Score in descending order

DOC TYPE VIEW FULL PAGE VIEW CITATION

PER



Effects of simultaneous switching noise on the tapered buffer design

Vemuru, S.R.

Very Large Scale Integration (VLSI) Systems, IEEE Transactions on

Published: Sept. 1997

Volume: 5 3 , Page(s): 290 -300



IEEE/IEE ELECTRONIC LIBRARY

HOME SEARCH ADVANCED SEARCH WHAT'S NEW



Refine Query:




(semiconductor or memory) and scan and (pcb or printed <near>circuit)

Search

Help

Collections: Journals Conferences StandardsYour search matched **14** of **506922** documents.**10** are presented on this page, sorted by Score in descending order

DOC TYPE	VIEW FULL PAGE	VIEW CITATION
CNF		<u>Using boundary scan test to test random access memory clusters</u> <i>Muris, M.; Biewenga, A.</i> Test Conference, 1993. Proceedings., International Published: 1993 , Page(s): 174 -179
CNF		<u>Scan test architectures for digital board testers</u> <i>Fichtenbaum, M.L.; Robinson, G.D.</i> Test Conference, 1990. Proceedings., International Published: 1990 , Page(s): 304 -310
CNF		<u>Testing conventional logic and memory clusters using boundary scan devices as virtual ATE channels</u> <i>Hansen, P.</i> Test Conference, 1989. Proceedings. Meeting the Tests of Time., International Published: 1989 , Page(s): 166 -173
CNF		<u>IEE Colloquium on 'Application and Development of the Boundary-Scan Standard' (Digest No.183)</u> Application and Development of the Boundary-Scan Standard, IEE Colloquium on Published: 1990
CNF		<u>Self-testable video controllers</u> <i>Sebaa, L.; Ladick, A.; Lofgren, K.; Cater, D.</i> WESCON/94. Idea/Microelectronics. Conference Record Published: 1994 , Page(s): 542 -546
CNF		<u>Checking signatures on boundary scan boards</u> <i>Lubaszewski, M.; Castro Alves, V.; Nicolaidis, M.; Courtois, B.</i> European Test Conference, 1993. Proceedings of ETC 93., Third Published: 1993 , Page(s): 339 -348
CNF		<u>International Test Conference 1989. Proceedings. Meeting the Tests of Time (Cat. No.89CH2742-5)</u> Test Conference, 1989. Proceedings. Meeting the Tests of Time., International Published: 1989
CNF		<u>International Test Conference 1988 Proceedings - New Frontiers in Testing (Cat.</u>

-  **No.88CH2610-4)**
Test Conference, 1988. Proceedings. New Frontiers in Testing, International
Published: 1988
- CNF  **Embedding virtual test points in PCBs**
Chenoweth, D.; Muegge, M.
WESCON/94. Idea/Microelectronics. Conference Record
Published: 1994 , Page(s): 529 -533
- CNF  **Built-in system test and fault location**
McLeod, G.R.
Test Conference, 1994. Proceedings., International
Published: 1994 , Page(s): 291 -299



IEEE/IEE ELECTRONIC LIBRARY

HOME SEARCH ADVANCED SEARCH WHAT'S NEW



Refine Query:

(semiconductor or memory) and scan and (pcb or printed <near>circuit)







Search

Help

Collections: Journals Conferences Standards

Your search matched 6 of 506922 documents.

6 are presented on this page, sorted by Score in descending order

DOC TYPE	VIEW FULL PAGE	VIEW CITATION
CNF		<u>Built-in system test and fault location</u> <i>McLeod, G.R.</i> Test Conference, 1994. Proceedings., International Published: 1994 , Page(s): 291 -299
CNF		<u>A modern PC-based sidescan sonar acquisition and display system</u> <i>Clapp, D.L.</i> OCEANS '93. Engineering in Harmony with Ocean. Proceedings Published: 1993 , Page(s): I402 vol.1
CNF		<u>A vertically integrated test methodology based on JTAG IEEE 1149.1 Standard Interface</u> <i>Ruparel, K.N.; Chin, C.; Fitzgerald, J.</i> ASIC Conference and Exhibit, 1991. Proceedings., Fourth Annual IEEE International Published: 1991 , Page(s): P11 -4/1-4
CNF		<u>Scan test architectures for digital board testers</u> <i>Fichtenbaum, M.L.; Robinson, G.D.</i> Test Conference, 1990. Proceedings., International Published: 1990 , Page(s): 304 -310
CNF		<u>Testability features of the 68040</u> <i>Gallup, M.G.; Ledbetter, W., Jr.; McGarity, R.; McMahan, S.; Scheuer, K.C.; Shepard, C.G.; Sood, L.</i> Test Conference, 1990. Proceedings., International Published: 1990 , Page(s): 749 -757
CNF		<u>International Test Conference 1988 Proceedings - New Frontiers in Testing (Cat. No.88CH2610-4)</u> Test Conference, 1988. Proceedings. New Frontiers in Testing, International Published: 1988

**CITATION and
ABSTRACT**

IEEE/IEE ELECTRONIC LIBRARY

HOME SEARCH ADVANCED SEARCH WHAT'S NEW

SEARCH RESULTS

VIEW FULL PAGE



A vertically integrated test methodology based on JTAG IEEE 1149.1 Standard Interface

- Ruparel, K.N.; Chin, C.; Fitzgerald, J.

Editor(s): Hsu, K.W., Tsai, Y.T., Parslow, P.

Vertex Semiconductor Corp., San Jose, CA, USA

This Paper Appears in :

ASIC Conference and Exhibit, 1991. Proceedings., Fourth Annual IEEE International

on Pages: P11 - 4/1-4

This Conference was Held : 23-27 Sept. 1991

1991

ISBN: 0-7803-0101-3

Total Pages: xxi+594

References Cited: 6

Accession Number: 4387324

Abstract:

A vertically integrated test methodology has been developed for ASIC testing based on the IEEE 1149.1 Standard Test Interface. A common interface is used to test at the wafer, packaged-chip and board/system levels. The boundary scan JTAG interface is combined with an internal full scan based test technique to provide a uniform test procedure at all stages of testing. At the prototype debug phase, the test circuitry is configured to test for design and process faults. At the manufacturing stage, it allows for efficient wafer sorting and packaged chip testing. At the board/system level, the same test set used at the wafer and package levels can be employed for incoming-inspection of parts and in-circuit-testing. In addition to basic scan testing, the protocol can perform AC/delay-fault testing. For embedded megacell and RAM module testing it is configured to control and operate an independent BIST scheme inside the ASIC device to achieve at-speed testing. This test methodology has been implemented on practical ASIC parts. The area overhead for the boundary scan architecture is on the order of a few percent for 30-50 K gate designs, and depending on the type of implementation, performance overhead varies from minimal to no penalty at the I/O cells. <>

Subject Terms:

wafer level; packaged chip level; board level; boundary scan JTAG interface; incoming inspection; in circuit testing; design faults; vertically integrated test methodology; JTAG IEEE 1149.1 Standard Interface; ASIC testing; internal full scan based test technique; process faults; wafer sorting; packaged chip testing; AC/delay-fault testing; RAM module testing; BIST scheme; at-speed testing; application specific integrated circuits; automatic test equipment; boundary scan testing; built-in self test; integrated circuit testing; printed circuit testing

Help

Copyright © 1997 IEEE -- All rights reserved

A VERTICALLY INTEGRATED TEST METHODOLOGY BASED ON JTAG IEEE 1149.1 STANDARD INTERFACE

Kamalesh N. Ruparel, Cary Chin and Jeff Fitzgerald

Vertex Semiconductor Corporation
1060 Rincon Circle
San Jose, CA 95131
Phone: (408) 456-9278 ext. 2342
FAX: (408) 456-9286

Abstract

A vertically integrated test methodology has been developed for ASIC testing based on the IEEE 1149.1 Standard Test Interface. A common interface is used to test at the wafer, packaged-chip and board/system levels. The Boundary Scan JTAG interface is combined with an internal full scan based test technique to provide a uniform test procedure at all stages of testing. At the prototype debug phase, the test circuitry is configured to test for design and process faults. At the manufacturing stage, it allows for efficient wafer sorting and packaged chip testing. At the board/system level, the same test set used at the wafer and package levels can be employed for Incoming-Inspection of parts and In-Circuit-Testing. In addition to basic scan testing, the protocol can perform AC/Delay-fault testing. For embedded megacell and RAM module testing it is configured to control and operate an independent BIST scheme inside the ASIC device to achieve At-Speed testing. This test methodology has been implemented on practical ASIC parts. The area overhead for the Boundary Scan architecture is on the order of a few percent for 30-50K gate designs, and depending on the type of implementation, performance overhead varies from minimal to no penalty at the I/O cells.

I. Introduction

This paper presents a vertically integrated test methodology developed for ASIC testing based on the IEEE 1149.1 Standard Test Interface. The JTAG test protocol is augmented to accomplish testing at the wafer, packaged chip and the board/system levels. Since JTAG is primarily a board-level test solution, in order to encompass a broader spectrum of test features, the JTAG instruction set was expanded and the internal test logic circuitry redesigned. The test methodology uses an internal full-scan based DFT technique and a robust I/O level testing procedure using the Boundary Scan Ring. In order to allow for Delay-fault testing, AC testing functions are included in the test protocol. For embedded megacells and RAM module testing, a Built-In Self Test (BIST) scheme is developed to provide for At-Speed testing and high fault coverage.

The following is a description of the test objectives & techniques used at all 3 phases of a chip's life cycle and the design/implementation of various chip-level test functions using the JTAG test protocol. Section II describes design enhancements to the protocol and the functional operation of scan test. Sections III, IV and V present the scan test procedures used at the wafer, package and board/system levels. The RAM BIST operations and megacell testing are described in Section VI. The practical results obtained over several implementations on various gate array families are discussed in Section VII. Finally, the conclusion is presented in Section VIII.

II. Enhanced Test Protocol

The Vertex ASIC design flow is based on a rigorous test methodology. The test protocol involves a full internal scan DFT methodology coupled with extensive features at the pin I/Os to perform a thorough wafer and package test sequence. An internally developed program, the Test-Logic-Insertion (TLI) software, automatically inserts the scan FFs in the core design so as to make the scan design process completely transparent to the customer. All Ver-

tex chips include a ring oscillator function, controlled by the test macro, formed out of the external and internal scan chains for performance characterization of the part. For megacells and RAM modules, a BIST structure is provided which is also controlled by the test macro. Additionally, a patented master-scan latch design that allows for delay-fault testing is also provided. All these features have been designed into the JTAG protocol in order to maintain a uniform test interface at all levels of chip testing. This was achieved by adding private instructions to the JTAG basic instruction set and enhancing the boundary scan cell design.

Instruction	Description
BYPASS	Required by 1149.1
EXTEST	Required by 1149.1
SAMPLE/PRELOAD	Required by 1149.1
INTTEST	Optional
IDCODE	Optional
INTSCAN(M)	Chip-on-Board Internal Scan
INPLACEIN	Chip-on-Board BSR Scanin
INPLACEOUT	Chip-on-Board BSR Scanout
WFRINTSCAN(M)	Wafer Internal Scan
WFRSCANIN	Wafer BSR Scanin
WFRSCANOUT	Wafer BSR Scanout
INTSCANP(M)	Package Internal Scan
INTROSC	Internal Scan Ring Oscillator
EXTROSC	External Scan Ring Oscillator

Table 1: Vertex Test Instructions

A total of 12 private instructions (9 + 3 for Master Scan), as shown in Table 1 were added. The instruction register size was expanded to 5 bits. (It is assumed that the readers are familiar with the operation and functionality of the JTAG protocol and hence no attempt has been made to describe the logic details of the protocol. Please refer to the JTAG spec for details [3].) For purposes of internal scan testing, a set of 3 individual instructions are required at the wafer and chip-on-board levels. At the wafer level, these are WFRSCANIN, WFRINTSCAN(M) and WFRSCANOUT. At the board level, these are INPLACEIN, INTSCAN(M) and INPLACEOUT. The requirement for internal scan testing dictates the use of only the *update* function during the external ring scanin and the use of only the *capture* function during the external ringscanout. Since the TAP controller state diagram requires passing through both the *capture-DR* and *update-DR* during any single TDR sequence, separate instructions were devised to handle the *capture* and *update* functions in different sequences. The difference between wafer scan test and chip-on-board scan test instructions is as follows: during wafer test, the signal is driven and observed through the I/O pads thereby providing a rigorous coverage of the I/O drivers, buffers and their control signals; during chip-on-board scan test, the signal is driven into the chip directly from the *update* stage and captured directly into the *capture* stage of the boundary scan cell, thereby bypassing the I/O area completely. Figure 1 shows the logic diagram of a generic Vertex Boundary Scan cell. At the package level, since the tester provides a broadside stimulus to the primary inputs and outputs, only one instruction, INTSCANP(M), is required to achieve internal scan testing.

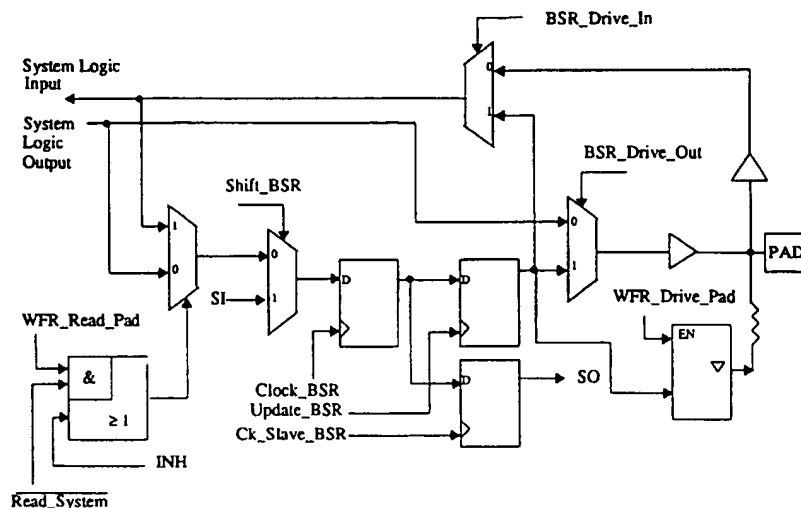


Figure 1: Generic Vertex Boundary Scan Cell

In addition to scan testing, the three internal scan instructions, WFRINTSCAN(M), INTSCAN(M) and INTSCANP(M), are also used to control the BIST circuitry for megacells and RAM modules in the Run-Test/Idle state. The INTROSC and EXTROSC instructions are used to configure the scan chains into ring oscillator functions for performance characterization.

III. Wafer Testing/Sorting

The scan test is essentially achieved by use of the 3 wafer scan test instructions as mentioned in the previous section. At WFRSCANIN, the stimulus loaded into the update stage of boundary scan cells is driven through the pad by use of the *wfr_drive_pad* signal shown in Figure 1. The cell configuration allows for the stimulus to be received by the core logic directly from the pad. The control is held in this state until the TAP controller is brought to the Run-Test/Idle state in the WFRSCANOUT instruction. After WFRSCANIN, the internal scan ring is loaded using the WFRINTSCAN(M) instruction. This is followed by the WFRSCANOUT instruction when the *wfr_read_pad* signal is activated to capture the value on the pad being driven out of the core logic due to the vector loaded into the internal scan chain. The capture takes place in the *capture-DR* state, by which time the *wfr_drive_pad* signal has long been turned off. The TAP controller is then shifted into the *shift-DR* state and the response vector scanned out. After returning to the Run-Test/Idle state, a system clock pulse is applied to capture the test response in the internal scan chain. The sequence is repeated for all the scan test vectors.

A special test sequence, *dynamic clocking*, has been devised to detect the faults at the tri-statable I/O drivers and their control signals for Bi-directional and 3-state output pins at wafer test. The technique of *dynamic clocking* has been incorporated into the ATPG tool developed in-house at Vertex. In this test sequence, during tri-state testing, the I/O pad is driven to a certain value at WFRSCANIN through a weak driver, specially designed for this purpose. Since the output driver is in the tri-state mode, the same logic value should be captured during WFRSCANOUT for the good machine. A different value captured will detect the faulty operation of the tri-state function on the output driver. The output driver function is tested by charging the pad with an opposite value through the weak driver at WFRSCANIN. At WFRSCANOUT, the value captured for a good machine should reveal the value driven by core logic through the output driver. In the addition to the regular scan testing, the wafer test procedure also involves testing the internal

and external scan chains in ring oscillator modes for performance characterization. A small parametric sampling is also done at this time. For instance, input leakage, input threshold and output drive test are also performed.

In the manufacturing environment, the benefits of a low pin count test interface are numerous. For example, a typical probe card consisting of approximately 256 probes can easily exceed several thousands of dollars. With this test methodology, the probe card costs are significantly reduced to below a few hundred dollars. The cost savings are also quite evident for repair and planarization of the probe cards. Another advantage of the low pin count interface is the ease of alignment, which can account for significant savings in test time overhead. With all pads probed, pad damage can be incurred due to repetitive probing making the packaging bond process far more costly. Other benefits lie in the cabling interface. By reducing the quantity of lines the mechanical interface is far more robust. Cable impedance qualification time (time-domain-reflectometry techniques) is also reduced thereby guaranteeing signal integrity to the DUT.

IV. Package Testing

At the package level, scan testing is accomplished with the use of only one instruction: INTSCANP(M). During package test, all boundary scan cells except the *inhibit* (enable) cell operate in a transparent mode. The major difference between the INTSCANP(M) and its wafer and chip-on-board counterparts [INTSCAN(M) & WFRINTSCAN(M)] is that in the latter cases, ATPG determines the value of the *inhibit* cell. For package test, the value in the *inhibit* cell is set manually and used to avoid conflicts at Bi-directional I/O cells that switch their direction between consecutive patterns.

Initially, the SAMPLE/PRELOAD instruction is used to load a dummy pattern with specified values at the *inhibit* cells. A dummy pattern is then applied by the tester at the device. The INTSCANP(M) instruction is then loaded and a test pattern scanned into the internal scan chain. The TAP controller is moved to the Run-Test/Idle state. In this state, a test pattern through the tester is applied at the primary inputs and outputs of the device, followed by a system clock sequence. The sequence from application of a dummy pattern to system clock application is then repeated for each scan vector. The dummy pattern is applied to protect the device from: (a) conflicts at the I/O pin due to a direction switch of a Bi-directional I/O cell caused by the change in core logic due to the system clock pulse, and/or (b) stray currents that may occur during the

internal scan sequence.

The boundary scan architecture provides for several other types of testing not otherwise easily possible at the device level. By using the *update* stage of the boundary scan cell, it is possible to test for power and ground bounce severities for several outputs switching simultaneously. Furthermore, many standard tests, such as continuity tests for opens/shorts and leakage tests can be performed by simply setting the appropriate values in the boundary scan ring without the need for an ATPG tool to figure out a correct pattern to be applied at the input of the combinational island that drives the primary I/O pins. Due to the simplicity of the interface, wafer sort tests can be cross correlated in the package environment.

V. Board/System Testing

The JTAG standard [3] specifies only the *INTEST* instruction to perform a broadside functional test for chip-on-board applications so as not to disturb other parts on the board. The *INTEST* instruction requires a special multiplexer (between the I/O pin and the core logic) that, in some implementations, can cause unacceptable performance penalties. The scan test instructions *INPLACEIN* and *INPLACEOUT* also require the presence of this multiplexer to allow internal scan testing of the device on board. However, the same set of test vectors that are used to test and qualify a part at the wafer and package levels is used at the board level in order to maintain a vertically integrated test philosophy.

Such a philosophy enables Incoming-Inspection of parts which could be essentially package level testing at the customer site. As mentioned in section III, the test signals do not flow through the I/O pad and I/O drivers at the board scan test level. Therefore, there is some loss of coverage over the faults on the control and data lines of the I/O drivers and buffers. However, due to the *EXTEST* instruction, this loss is compensated by any board level interconnection test sequence that would test for these faults.

The scan test sequence at this level is identical to that at the wafer level except for the use of different instructions. The initial boundary scan ring is loaded with the *INPLACEIN* instruction which only allows an *update* operation and no action takes place in the *capture-DR* state. The *INTSCAN(M)* instruction is then used to load the internal scan chain. The primary outputs' response is then *captured* using the *INPLACEOUT* instruction in the *capture-DR* state with no action taking place in the *update-DR* state. This is followed by a system clock pulse to capture responses in the internal scan chain. The procedure is repeated for each scan vector for the device.

In addition to the above, the protocol offers all the advantages of rigorous testing at the board level, such as interconnection testing, glue-logic island testing, etc. for which it is designed. Moreover, the test methodology used for the board level is fully applicable to an Multi-Chip Module (MCM) scenario.

VI. Megacell/RAM and AC/Delay-fault Testing

A specially developed automation software generates BIST circuitry around the periphery of a megacell or RAM module. The BIST logic control is part of the internal scan chain, thereby enabling full control of its operation through the internal scan chain and the test macro. The BIST logic is operated in the Run-Test/Idle state of the internal scan instructions: *INTSCAN(M)*, *INTSCANP(M)* and *WFRINTSCAN(M)*. The control signals *jst* and *jby* are provided through two extra scan FFs located in the BIST logic. In the functional operation mode, these FFs default to values that allow the RAM to operate in the normal mode. During regular scan test, the RAM is put in a bypass mode and during RAM test, appropriate values in these FFs allow independent module testing. The use of special FFs to control the test operation for each RAM module allows flexibility to perform individual module debug

as well as parallel testing of several modules simultaneously. For RAM modules, the 13N algorithm [1] is used to generate the test vectors. It has been shown [2] that this algorithm detects all stuck-at, transition, state-coupling, multiple-access and stuck-open faults. The BIST logic is operated to achieve At-Speed testing of the RAM modules.

For AC performance characterization, the *INTROSC* and *EXTROSC* instructions configure the internal and external scan chains, respectively, into ring oscillators. Since the internal scan ring consists of several thousands of latches, the results obtained from the ring oscillator measurements provide excellent correlation to actual performance of the parts. For delay-fault testing, a special patented master-scan latch design is used to control the operation of a smooth 1-to-0 and 0-to-1 transition capture and measurement. (The discussion of the design and operation of the master-scan latch is beyond the scope of this paper.)

VII. Results

The test methodology presented in this paper has been implemented on practical ASIC parts. Tables 2 & 3 show various area overheads over two generations and different sizes of Vertex gate arrays. In the V25 series, the JTAG boundary scan cells were designed and placed in the core logic area. As can be seen from the table, this proved to be a costly process for lower than 10K available-gate designs. Due to the *INTEST* performance penalty being more than a nanosecond, the V25 series offers both variants of boundary scan cells: with and without *INTEST*. In the V50 series, this was optimized to include the boundary scan logic in the I/O cell area, reducing the performance penalty to less than 0.1 nanoseconds. This is because, in the V50 series, a novel circuit design with custom layout was implemented to place the multiplexer before the input buffer/driver. It can be seen from the table that the JTAG area overhead incurred for the V50 gate arrays is only due to the test macro, which consists of only a few hundred gates. Realistic assumptions were made over the number of functional I/Os and RAM modules used for each gate-array size. For RAM modules, the following assumptions were made: 1 2K-bit RAM module for less than 30K used-gates design, 2 modules for designs between 30K and 100K used-gates and 3 modules for designs greater than 100K used-gates. The area overhead for the JTAG boundary scan cells makes a significant impact (on the order of few percent) when a comparison between the V25 and V50 series is made.

Figures 2 and 3 show the area overhead plots for the two families V25 and V50 with respect to the impact of size of gate array (used gates) and the type of routing technology used (double-level-metal vs. triple-level-metal). The overhead numbers include both JTAG and internal scan penalties. As the number of used gates increases, the impact of scan and JTAG overheads decrease significantly. The same trend is seen for routing technologies owing to an increase in usable gates for each gate-array size for the triple-level-metal scenario.

Since Vertex uses a full internal scan methodology, all parts are tested for greater than 99% stuck-at fault coverage. The test vectors are generated from a powerful ATPG tool developed internally and well-proven through years of refinement over the course of many designs. A patented design allows for zero performance penalty due to the scan latch. Due to an optimized physical design/layout flow, which entails a post-layout scan chain connection and rigorous timing analysis/optimization, the performance penalty due to scan chain routing is negligible. The test pin overhead is only the 4 (+1 optional) pins required for the JTAG interface.

VIII. Conclusion

A vertically integrated test methodology based on the standard JTAG test interface has been developed and presented in this paper. The protocol was implemented on two generations of

V25 Family Size	Utilized Gates	Functional I/O	Test Macro Overhead	BS Cell Overhead	Internal Scan Cell Overhead	No. of 2K-bit RAM Modules	BIST Logic Overhead	Total Overhead	%age Overhead
37K	25900	168	700	1512	1701	1	1017	4930	19%
54K	37800	204	700	1836	2847	1	1017	6400	17%
89K	62300	260	700	2340	4062	2	2034	9136	15%
129K	90300	312	700	2808	7305	2	2034	12847	14%
172K	120400	360	700	3240	9598	2	3051	16589	14%

Table 2: Area Overhead over Used Gates for the V25 Gate-Array family with RAM

V50 Family Size	Utilized Gates	Functional I/O	Test Macro Overhead	BS Cell Overhead	Internal Scan Cell Overhead	No. of 2K-bit RAM Modules	BIST Logic Overhead	Total Overhead	%age Overhead
24K	16800	120	700	0	950	1	1017	2667	16%
30K	21000	168	700	0	1366	1	1017	3083	15%
41K	28700	204	700	0	2128	1	1017	3845	13%
54K	37800	204	700	0	2385	2	2034	5119	14%
86K	60200	260	700	0	4603	2	2034	7337	12%
107K	74900	312	700	0	6058	2	2034	8792	12%
146K	102000	360	700	0	8097	3	3051	11848	12%
198K	138600	360	700	0	11720	3	3051	15471	11%
256K	179200	360	700	0	15740	3	3051	19490	11%

Table 3: Area Overhead over Used Gates for the V50 Gate-Array family with RAM

gate-array families: V25 and V50. It was shown that by incorporating the JTAG boundary scan cells in the I/O area significant advantages in area overhead and performance were obtained. The results presented from realistic experiments over a wide variety of gate-array sizes and different routing technologies show that complete device testing at all levels can be achieved with very reasonable silicon overhead and minimal performance impact.

The test methodology accomplishes testing at all levels ranging from the wafer level up to the board and system levels. At the prototype debug stage, test circuitry and functions are provided to achieve efficient testing of design and process faults. At the manufacturing stage, the procedures provide for quick wafer sorting without the need to probe all pins and efficient package testing using internal full scan based test techniques. A uniform test pattern set that is used at the wafer and package phases can also be used for Incoming-Inspection of parts and chip-on-board scan test applications. The test methodology demonstrates a powerful and standard way to achieve most of the ASIC testing objectives with acceptable area and performance overheads.

Acknowledgements

The authors would like to thank Brent Miller for his contribution to the initial design of this test methodology, Jun Zhu for helping out in the verification of the design and Jwu-Rong Peng for her help in determining the circuit and performance issues with the boundary scan cell implementation.

References

- [1] Dekker, R., Beenker, F., and Thijssen, L., "A Realistic Self-Test Machine for Static Random Access Memories," International Test Conference, pp. 353-361, September 1988.
- [2] Dekker, R., Beenker, F., and Thijssen, L., "Fault Modeling and Test Algorithm Development for Static Random Access Memories," International Test Conference, pp. 343-352, September 1988.
- [3] *IEEE Standard Test Access Port and Boundary-Scan Architecture*, May 1990.
- [4] Eichelberger, E.B., and T.W. Williams, "A Logic Design Structure for LSI Testability," Proc. 14th Design Automation Conference, pp. 462-468, 1977.
- [5] McCluskey, E.J., "IC Quality and Test Transparency," International Test Conference, pp. 295-301, September 1988.
- [6] Ruparel, K. and Zhu, J., *JTAG Boundary Scan Test Interface for the Vertex V25 Gate Array Family*, April 1991.

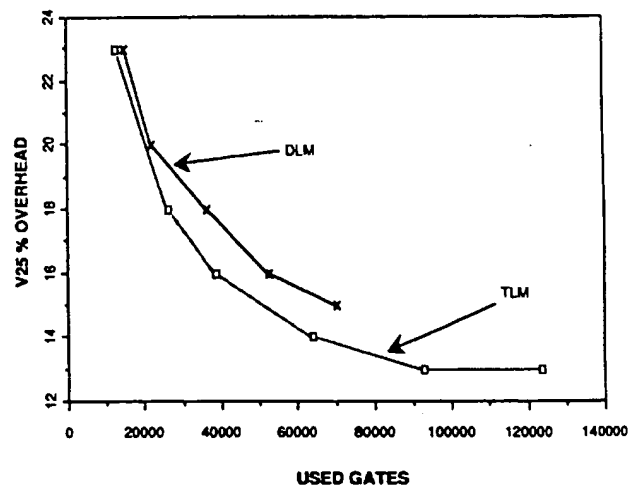


Figure 2: Area Overhead Plots for the V25 family with double and triple level metal without RAM

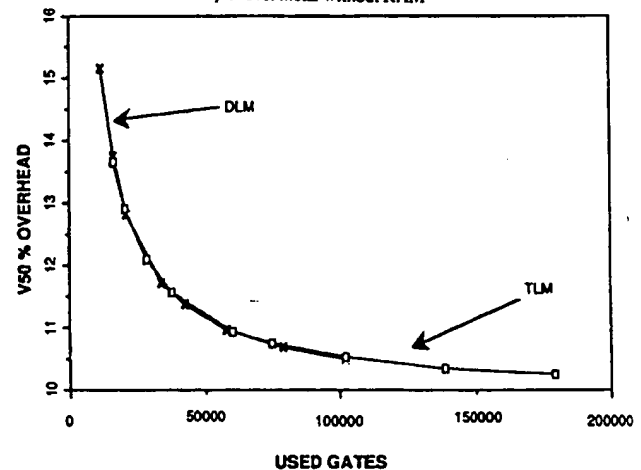


Figure 3: Area Overhead Plots for the V50 family with double and triple level metal without RAM

CITATION and
ABSTRACT

IEEE/IEE ELECTRONIC LIBRARY

HOME SEARCH ADVANCED SEARCH WHAT'S NEW

SEARCH RESULTS

VIEW FULL PAGE



Testability features of the 68040

- Gallup, M.G.; Ledbetter, W., Jr.; McGarity, R.; McMahan, S.; Scheuer, K.C.; Shepard, C.G.; Sood, L.

Motorola Inc., Austin, TX, USA

This Paper Appears in :
Test Conference, 1990. Proceedings., International

on Pages: 749 - 757

This Conference was Held : 10-14 Sept. 1990
1990 ISBN: 0-8186-9064-X

Total Pages: xvi+1083

References Cited: 11

Accession Number: 3985687

Abstract:

The design and implementation of on-chip test functions on the 68040 microprocessor are described. The discussion includes an introduction to the 68040, along with the testability goals and objectives that were set at the beginning of the design. Further discussions detail the different design-for-testability techniques used to control and observe the behavior of the 68040 subsystems. Topics covered include the global test architecture, special test modes for the internal RAM arrays, the scan circuitry used for structural testing of random logic, and the IEEE 1149.1 (JTAG) implementation on the 68040. ◊

Subject Terms:

IEEE 1149.1 JTAG; cache test; scan test; on-chip test functions; 68040 microprocessor; testability; global test architecture; internal RAM arrays; scan circuitry; structural testing; random logic; computer architecture; integrated circuit testing; integrated memory circuits; logic arrays; logic testing; microprocessor chips; printed circuit testing; production testing; random-access storage

Help

Copyright © 1997 IEEE -- All rights reserved

Testability Features of the 68040

Michael G. Gallup, William Ledbetter Jr., Ralph McGarity, Steve McMahan,
Kenneth C. Scheuer, Clark G. Shepard, and Lal Sood

Motorola, Inc.
Microprocessor Products Group (OE38)
6501 Wm. Cannon Dr. West
Austin, Texas 78735-8598

Abstract

This paper describes the design and implementation of on-chip test functions on the 68040 microprocessor. The discussion includes an introduction to the 68040 along with the testability goals and objectives that were set in the beginning of the design. Further discussions detail the different design for testability (DFT) techniques used to control and observe the behavior of the 68040 subsystems. Topics covered include the global test architecture, special test modes for the internal RAM arrays, the scan circuitry used for structural testing of random logic, and the IEEE 1149.1 (JTAG) implementation on the 68040.

INTRODUCTION

The 68040 is a third generation 32-bit microprocessor that executes the 68000 family instruction set [1] [2]. As shown in Figure 1, the 68040 includes an integer processing unit (IU), an IEEE 754 compatible floating point unit (FPU), and a memory unit comprised of separate 4K byte instruction and data caches and separate 64-entry address translation caches (ATCs). Table 1 shows statistics and features of the 68040 and Figure 2 shows a die photo.

Die Size	14.4mm x 15.5mm
Transistors	1.2 million
Process	0.8μ
Operating Frequency	25 MHz
Operating Voltage	5.0V +/- 5%
Package	179 pin PGA

Table 1 - 68040 Statistics

Testability Goals

It was realized early in the project that a purely functional approach to testing the 68040 was impractical from a time and test complexity standpoint. It was also realized that a functional approach would likely result in lower than de-

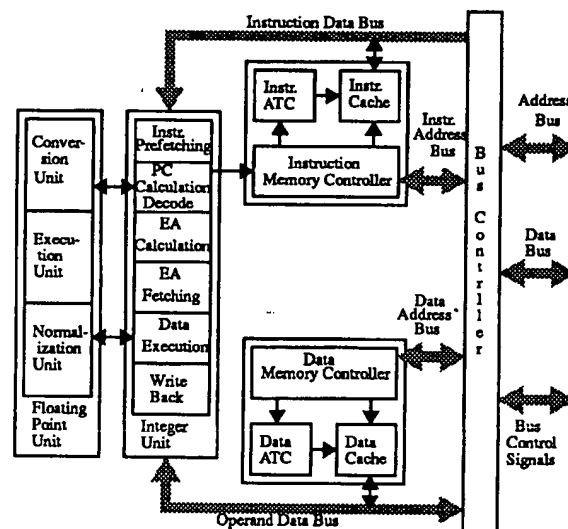


Figure 1 - 68040 Block Diagram

sired test coverage. This led to the development of separate DFT methodologies and techniques that best suited each section of the design. To address customer's board test requirements, the IEEE 1149.1 standard test interface was included.

The following testability goals were defined:

1. Minimize the number of manually generated test patterns. The fault coverage for the test patterns had to be as high as possible. This meant that the DFT techniques must provide good controllability and observability.
2. The silicon area and verification costs of each DFT technique must be reasonable. Functionality of the DFT methods must be verifiable before the design is fabricated.

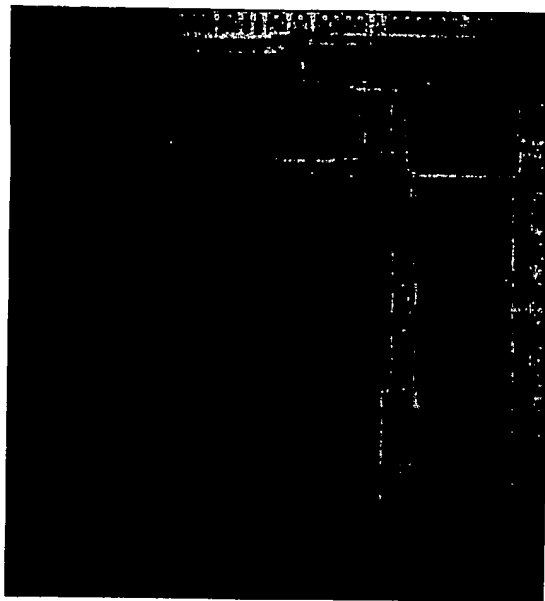


Figure 2 - 68040 Die Photograph

Test Architecture

The 68040 test architecture utilizes three different DFT methods. A functional approach was taken for all data paths in the IU and FPU as well as the data paths to the external I/O pins. These paths are tested with normal-mode instruction sequences.

Ad-hoc test modes were constructed for testing of the on-chip memory arrays and the on-chip phase-locked loop (PLL). The cache data and tag arrays employ simple extensions to the existing functional snoop logic to allow the arrays to be accessed from the bus in a manner similar to a static RAM array. Micro-code sequences were constructed in the IU to perform MARCH test primitives on the data and instruction ATCs.

The ROM/PLA and control areas use a structured custom scan approach. Scan tests are generated by automatic test pattern generation (ATPG) software developed in-house [3].

Figure 3 shows the amount of die area and transistors that are covered by each of the DFT methods. From this figure it can be seen that the first goal of the DFT philosophy has been accomplished: the number of functional test patterns has been reduced.

Global Test Interface

Coordination of the test logic is accomplished with a global test bus which is driven from two 5-bit registers. Included in this bus are two master mode signals which indicate one of three major test modes the 68040 may be in: normal mode where the 68040 functions as a microprocessor, ad-

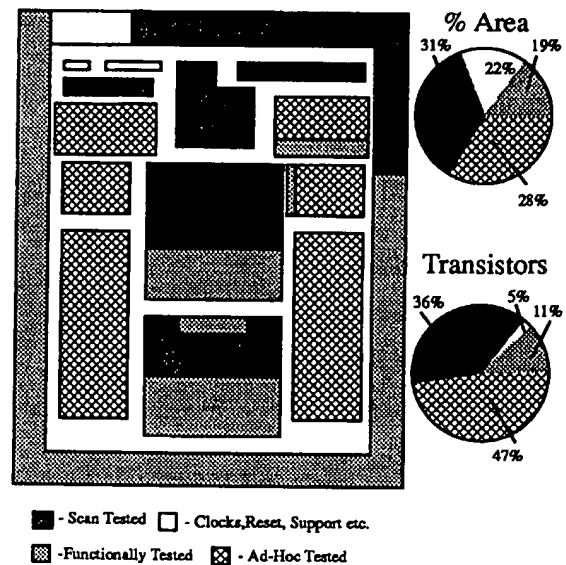


Figure 3 - DFT Method coverage

hoc mode for testing of the on-chip memory arrays and the PLL, and scan test mode for testing random logic and ROM/PLA arrays. Each of the 68040 subsystems monitors the global test bus for the indication that it is selected to perform its required task.

Ad-hoc modes use only one of the registers. The register value is used to select cache data, cache tag, ATC or PLL ad-hoc test functions. Cache data and tag modes, for example, place the external bus controller and the two internal memory controllers into states that allow test functions to operate on the arrays.

Scan test mode uses both test registers. The ten bits on the global test bus are broken down into three fields. The first field (3 bits) is used to select the major subsystems, like the IU or the FPU. The second field (5 bits) is used to select the desired scan chain in the subsystem. By using these two fields, each scan chain has a unique test address. The last two bits control the state of the scan chains. There are three possible states that a scan chain can be in: idle, capture, and shift. The idle state is one in which the clocks to the scan chain are off: both scan shifting and data capture are disabled. The capture state is used for capturing data from the logic that feeds the scan chain. The shift state is one in which data is being either shifted into or out of the scan chain.

Data to be loaded into a scan chain is first loaded into a 32-bit parallel-to-serial shift register located in the data pad logic. This shift register is also used as part of the 1149.1

logic for the boundary scan data register, which minimizes scan test silicon. Once the data is loaded into the data pad register, the shift state is entered and the data is shifted into the addressed scan chain that is pointed to by the values in the global test registers. This procedure is repeated until all of the input scan chains to a given logic block are loaded. The capture scan chain is addressed and the capture mode is entered. After a given number of clocks, the value of the output scan chain is shifted back into the data pad shift register and driven out to be observed on the data pads. This is how scan test patterns are applied to the internal scan tested portions of the 68040.

Ad-Hoc Test Modes

Ad-hoc modes are used on subsystems where scan or functional testing would be too costly in time, test complexity or silicon area. These subsystems are the cache data and tag arrays, the ATCs, and the PLL module.

PLL Ad-Hoc Mode

The 68040 utilizes a phase-locked loop (PLL) to generate the internal clocks [4]. The PLL module controls the skew between the external system bus clock (*BCLK*) and the internal clocks, which are derived from the processor clock (*PCLK*). The skew control is performed with all digital circuitry.

As shown in Figure 4, the PLL is comprised of four major blocks: the phase detector, the linear up/down shifter, the delay line and the clock generator. Two test mode functions

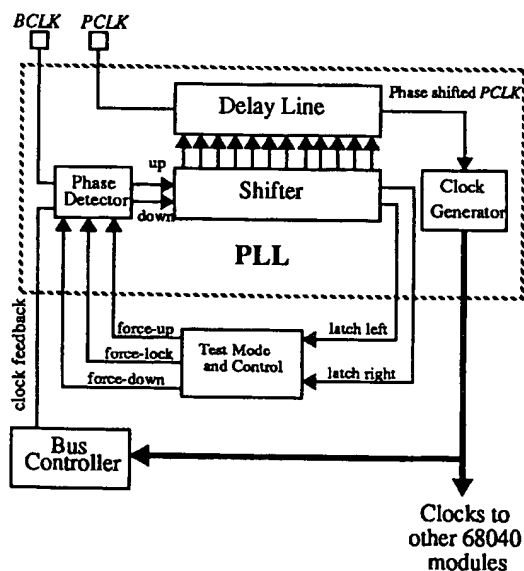


Figure 4 - PLL Block Diagram

were incorporated to insure testability of these blocks. The first mode allows the phase detector output to be overridden and forces an increase or decrease in delay between incoming *PCLK* and internal *PCLK*. This mode provides the capability to deterministically force the linear shifter and delay line to select each shift cell and delay line inverter pair. Correct operation of each delay line inverter pair is guaranteed by the observation of the following shift of the linear shifter, since the shift control logic requires the internal *PCLK* to toggle in order to shift.

The second test mode function addresses observability of the linear shifter. Latches were placed at each end of the linear shifter. The outputs of these latches are observable on I/O pads during test mode. This observability, in conjunction with the controllability provided by the phase detector override, allows a straightforward test algorithm which insures that each shift cell and each delay line inverter pair have no stuck-at faults. Phase detector functionality is verified in the same manner. A normal *PCLK* input with a constant high *BCLK* input should result in a phase detector shift up indication and a constant low *BCLK* input should result in a shift down indication. Observation of the latches confirms the proper operation of the phase detector.

Cache Data and Tag Ad-hoc Mode

The caches integrated on the 68040 are highly embedded and have limited observability and controllability from a functional standpoint. In the past, cache arrays were typically small and a functional data path test approach controlled by microcode was utilized [5] [6].

With the cache arrays accounting for a large percentage of the 68040 die area, it was imperative to define the test objectives for these arrays before design began. It was determined that a functional test approach would require several million test vectors to be generated, without an adequate characterization facility.

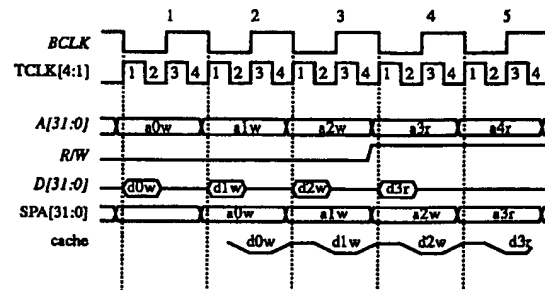
Characterization was deemed mandatory for the 68040 design. The characterization process helps to identify design and process related weaknesses. Since cache arrays are fabricated with the tightest feature and spacing rules available in a given technology, process-related defects in the arrays tend to show up earlier in the manufacturing cycle. This presents itself as an opportunity for yield enhancement by classifying and understanding these defects. Characterization (shmooing and 'bit mapping') is an integral part of test and yield improvement activities. The large die size of the 68040 makes yield improvements crucial. All of these factors led us away from the 'functional' test approach.

The major difficulty inherent in cache array testing is identical to that of testing any memory structure: the sub-structures that make up the entire array (row decoders, column

The 68040 has separate data and instruction caches (Dcache & Icache) and their associated tag memory arrays and control logic. Test logic was added so that these arrays can be partitioned to appear from the external bus as stand-alone memory. Since these arrays are functionally designed as 4-way set-associative caches, additional logic was added to bypass the internal cache and tag replacement counter in the ad-hoc test mode. This enabled us to test the four sub-arrays independently.

normal operation: test mode operation simply adds control to operate the data paths differently. The Icache is accessed via the index address lines A[9:4] and the longword (LW) select lines A[3:2], as shown. The index enables one of sixty four row decoders, and the LW select picks 1-of-4 LWs from the line. In the normal mode of operation, the address lines are driven by the IU, but in test mode these address lines are driven by the tester through the snoop address path. The data to the cache array is supplied by the tester through the line read buffer. This enables the tester to generate and supply address dependent data patterns for the arrays. During the ad-hoc read cycle, 1/2 line of data (64 bits) is read from the cache into the holding register (CLHR), as shown. The data is then placed onto the internal 32-bit data bus which is connected to the data pads.

Paper 33.2
752



dress pins and the data to be written to the data pins. The *R/W* line is set to indicate write. The line read buffer is loaded in *TCLK*[3] with the *LW* data and the snoop address buffer is loaded in *TCLK*[4] with the address. All four *LW*s in the read buffer are loaded with the same data, which constrains the data array test to a single *LW* in a line at a time. In the next *TCLK*[1] the addresses are received and decoded by the row decoders to select one row line on the cache data array. The following *TCLK*[2], the entire line in the cache is written with the data in the line read buffer.

Due to the pipelined nature of the access sequence to the data arrays, the transition from reads to writes causes the first write data to come directly from the line read buffer. This implies that the test sequence must provide a write cycle before the read sequence to preload the line read buffer. Then, the first write address given after the read sequence will write the data that was preloaded in the line read buffer.

Figure 7 shows the block diagram for the four way set associative data tag array (Dtag). The array is arranged in a 32 by 216 bit configuration. The bit line length was shortened to improve access time by folding the array on address $A[9]$. The array contains 22 data bits (the upper 22 address bits) plus 1 valid bit and 4 dirty bits for each set. Index addresses $A[9:4]$ are supplied by the snoop address buffer as

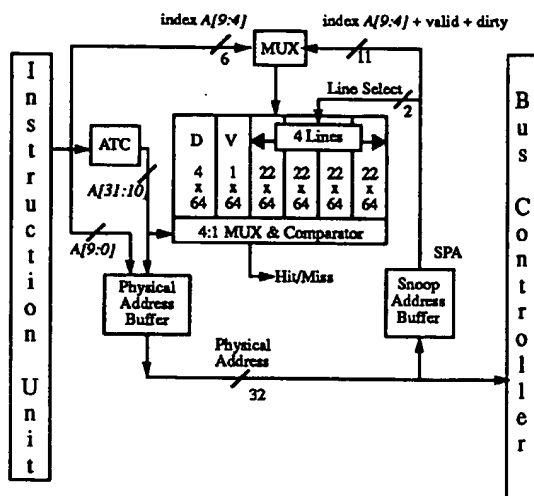


Figure 7 - Data Cache Tag Test Mode

shown. In test mode, one of 32 decoders are selected by the $A[8:4]$ address bits, and $A[9]$ selects 1 of 2 columns. The data to be written into the array is supplied through address pins $A[31:10]$ (22 bits), $UPA[0:1]$ and $TT[0:1]$ pins (dirty bits) and $CIOUT$ pin (valid bit). One of the four lines in the selected set is selected through the $TLN[0:1]$ pins. Thus, complete controllability and observability from the external I/O pins to the array is established. In this way the tester can exercise the tag array as a stand alone memory device.

Figure 8 shows the timing diagram for a read/write operation for the Dtag array. For an array read, the $A[9:4]$ index

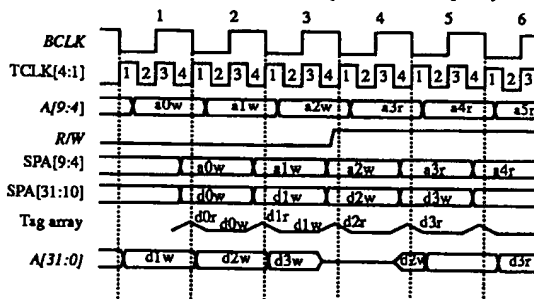


Figure 8 - Cache Tag Test Mode Timing

address is loaded in the snoop address buffer in $TCLK[4]$. For a write, $A[31:10]$ is also loaded as the tag data to be written. In $TCLK[1]$ the index is decoded by the row decoders. The data is written into the tag array in $TCLK[4]$ and is read out during $TCLK[2]/TCLK[3]$. The instruction tag array is tested in a similar manner.

The implementation of a robust test suite covering the two

types of embedded arrays (data and tag) on the 68040 has been handled in several ways. One of the major goals of the memory test design was to allow for sequence independent access, both of the data/address stream and of the sequences of reads and writes. Another was to be able to make use of algorithmic pattern generators (APG) that are available on VLSI testers to derive the address and data sequences to stimulate the arrays. This required two methods of access for reading or writing the arrays, due to the pipelined nature of the design.

The first method uses single bus clock primitives that are either a read or write cycle as described in the above text. This method requires that the data being compared in a read cycle be 'pipelined' from the address two reads back, while a write requires a few starting cycles, along with the correct preloaded data for the first write. The second method created two 'primitives', read-flat and write-flat, which consist of 3 reads or 3 writes to the part. The data observed from the flat-read is the result of the first read from the cell addressed, but two extra reads occur in order to get the data to the bus. The flat-write cycle either executes a dummy read followed by a write of the preloaded data (if the flat-write is just after a read) and then a write of the desired data to the desired cell, or three writes to the desired cell (if the flat-write is after a previous write cycle).

The vast majority of traditional memory tests [7] attempt to detect 3 classes of faults. The classes are: faults due to nodes that are stuck at one value, faults caused by an access transition from one cell to a different cell, or faults caused by coupling effects between cells. The second access method described above does not compromise the test coverage for algorithms generated to detect these types of faults, but does 'flatten' out the pipelined read and write accesses. This method makes the array look and act as a normal memory array, and standard memory test algorithms were very easy to implement on the APG. The flat method is not adequate for all memory test algorithms. Tests that require one and only one read or write to a given cell in the array have been implemented using the single bus clock primitives, and are driven by using straight line vector patterns, instead of relying on the APG.

Due to the implementation of the data path to the Icache, data written to one of the four banks (LW's) is copied into all four banks. This required the segmentation of the Icache test into 4 separate array tests. Test coverage to insure that the 4 banks are selected independently is achieved by functional testing.

ATC Ad-Hoc Mode

In order to test the on-chip data and instruction address translation caches (ATCs), test microcode was added to the IU. The ATCs are organized as 4-way set associative cach-

es. We desired to have the test sequence the same for both the data and instruction ATCs. Furthermore, we wanted to have a sequence that was invariant for both passing and failing conditions so that the tester would not get out of sequence after detecting the first failure. In order to achieve these goals the microcode loop in Figure 9 was used.

```

LOOP:
  Read LA1 information from d pins
  Translate LA1 and report results to d pins
  Read LA2 information from d pins
  Read PA information from d pins
  Load ATC entry using LA2 and PA information
GOTO LOOP

```

Figure 9 - ATC Test Microcode Flow

The LA1 (Logical Address 1) and LA2 information are obtained in one 32-bit access each and include the following: the upper bits of the logical address, a 2-bit indicator of which set should be loaded, a supervisor/user indicator, and an instruction/data ATC indicator.

The PA (Physical Address) information is also obtained in one 32-bit access and includes the upper bits of the physical address and associated status information such as a write protect bit and a resident bit. This is the information which is normally loaded into the ATC.

The instruction/data ATC indicator for LA1 controls which ATC attempts the translation. The instruction/data ATC indicator for LA2 controls which ATC gets loaded. Since only one ATC is tested at a time, the load or translate step can be effectively avoided by forcing the operation to occur to the ATC which is not being tested. Thus all four MARCH [7] operations can be performed.

The steps which read information from the I/O pins or report status to them do so by using the IU's normal data read and write capability, except that the ATCs are forced to be disabled. For example, to read LA1 information the IU supplies an address, requests a read, and obtains the needed information from the data pins. Similarly, the step which reports the results to the pins is performed by the IU by supplying an address and the data, and requesting a write access. Thus the IU used already existing capability to perform this subtask. The mechanisms used to control the Supervisor bit and the ones which controlled which ATC should be activated were also the same as the mechanisms used in normal mode. In fact, special test logic was only required for these capabilities:

1. forcing a specific set to be loaded (normally a random one is chosen)
2. the ability to directly read into the IU the results of the translation, including the PA and associated attributes

3. the ability to read into the IU an indication that the attempted translation hit or missed.

Note that when information is obtained from the data pins, the address supplied by the IU is arbitrary. But to ease the creation of the tester patterns, we chose to increment the address by four each access. This allowed us to create ATC test patterns using existing assembly language-based tools, since each input can be placed in a unique memory location and each output goes to a unique memory location.

The controllability and observability provided by the test microcode loop allowed us to write a compact 11n MARCH test for each ATC.

Scan Test

Random logic areas of the 68040 are tested with a custom scan methodology. In the initial phases of the design, an evaluation of level sensitive scan design LSSD [8] was made. Due to the practical requirements of a static design, and the silicon area involved, this method was not used. Instead, a custom scan method was derived which is similar to the Stanford scan path topology [9].

The areas that are scan tested include: control logic in the integer and floating point units, PLAs and ROMs, the data cache controller (DMEMC), the instruction cache controller (IMEMC) and the bus controller (BC). The latches in these areas are interconnected in chains to isolate the logic. Test patterns are shifted through the chains by the on-chip Factory Test Controller. An example of scan tested logic is shown in Figure 10. In this case, the inputs to the combinational logic under test (CLUT) fanout from two chains and the outputs terminate on one chain. The detail of part of a typical scan chain is shown in Figure 11. The scan latches

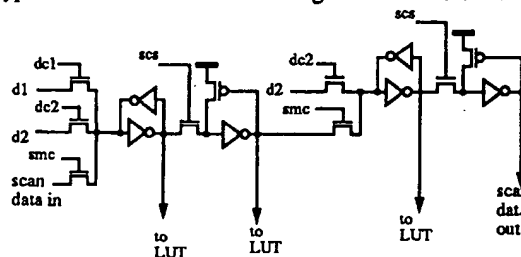


Figure 11 - Detail of a Typical Scan Chain

are master-slave latches, with the master latch also used in normal chip functions. In this case, the master latch is fed by two logic functions *d1*, *d2* which are normally clocked in on two different chip clocks *dc1*, *dc2*. The master clock *smc* loads data fed from the previous scan latch, and the slave clock *scs* transfers the data from the master to the slave.

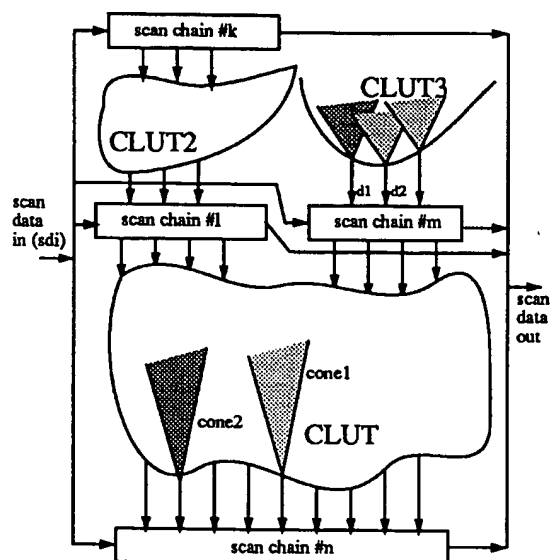


Figure 10 - Example of Scan-Tested Logic

The timing of a single scan test is shown in Figure 12. The

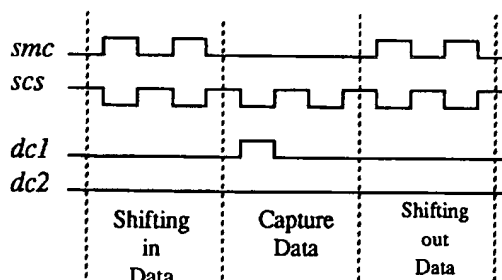


Figure 12 - Scan Chain Timing

master and slave clocks shift a pattern into each scan chain which is an input and/or output of the logic. One of the normal mode clocks to the master latch is pulsed to clock the test result into the output chains. The master and slave clocks then shift the result out of the chain. If the logic under test is a PLA or ROM, the test controller will pulse the appropriate normal clocks to cycle the structure and propagate the test to the output scan chains.

During factory test, the factory test controller assumes control of the 1149.1 boundary scan register. The interface of the scan chains to the factory test controller and to the I/O pins is shown in Figure 14. In factory test mode, the factory test controller has control over the master, the slave, and the

normal clocks for each scan chain. The inputs of every chain are wire-ORed and multiplexed from the 1149.1 boundary-scan ring at the end of the data pad inputs. Likewise, the outputs of the chains are wire-ORed and multiplexed into the beginning of the 1149.1 scan ring at the beginning of the data pad outputs. This allows the input and output vectors to be parallel loaded from/to the data bus using separate, contiguous sections of the 1149.1 scan ring. Figure 13 shows a summary of the scan chain statistics on the 68040.

Section	No. of scan chains	Total No. of bits	Maximum chain length
IMEMC	7	103	28
DMEMC	4	277	80
BC	1	115	115
IU	22	1145	128
FPU	14	750	118
Totals	48	2390	

Figure 13 - 68040 Scan Chain Statistics

IEEE 1149.1 User Test Port

We realized that due to the development of high density printed circuit boards (PCBs), some form of board test assistance was required on the 68040. The 68040 provides a user accessible circuit that is compatible with the IEEE 1149.1 (JTAG) standard test access port and boundary scan architecture [10] [11]. The 68040 implementation provides the capability of testing PCB interconnections independent of the normal chip function. The test logic on the 68040 consists of two test data registers, a 3-bit instruction register, 5 dedicated signal pins, and a 16 state test port controller (TAP). The two test data registers are the single-bit bypass register and the 184-bit boundary scan data registers. It is estimated that the 1149.1 logic comprises 0.3% of the total die area. This is an inexpensive investment for the benefits that this test circuit provides the PCB designer. Figure 14 details the functional blocks in the 1149.1 interface. This implementation supports the three required instructions (SAMPLE/PRELOAD, EXTEST, and BYPASS) but does not include any of the optional public instructions. The instruction register decodes five unique instructions: BYPASS, EXTEST, SAMPLE/PRELOAD, SHUTDOWN, and HI-Z.

The BYPASS instruction reduces the number of shift register bits between the *TDI* and *TDO* pins to a single cell when testing other devices on the 1149.1 circuit. The SAMPLE/PRELOAD instruction is used to capture a "snapshot" of the pins on the 68040 during normal operation. It can also be used to preload the boundary scan data register to an ini-

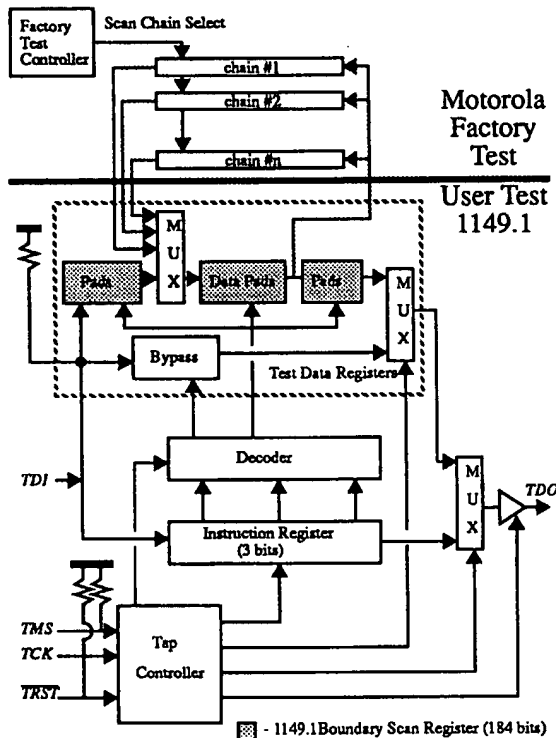


Figure 14 -68040 Test Interface

tial state. The EXTEST instruction selects the boundary scan data register. Five bits are provided in the boundary scan register to control the direction of the I/O signal pins. With the TAP controller, the user can load data to the output pins or capture data from the input pins. This way the boundary scan data registers can be used to test PCB traces, solder joints, socket connections and package interconnections.

To further assist the PCB designer, two additional instructions have been added: HI-Z and SHUTDOWN. The HI-Z instruction can be used to place all output-only pins, which normally do not have this control, into a high impedance state. This allows the user the capability of completely isolating the 68040 from the board. The SHUTDOWN instruction is provided to assist with the clocking restriction that the 68040 imposes. Due to the dynamic nature of the 68040, there is a requirement that the *PCLK* and *BCLK* clocks always be running. The SHUTDOWN instruction selects the bypass register and enables an on-chip oscillator giving complete freedom to test the *PCLK* and *BCLK* input pins. The EXTEST and HI-Z instructions also enable the on-chip oscillator clock function.

Results and Conclusion

An aggressive DFT philosophy was adopted so that the 68040 could be easily tested. The two main goals for testability were achieved by adopting the best test methodology for each part of the 68040. For the data paths in the IU and FPU sections, functional tests are used. For the large embedded memory arrays, ad-hoc test modes have been built which allow the arrays to be viewed as static memory. The random logic and ROM/PLA sections, which are traditionally the hardest sections to test, are tested via a scan test methodology. To provide the user with external board test functions, the IEEE 1149.1 interface is included. This interface provides the PCB designer with the capability to test the PCB for faults that would otherwise require complicated functional tests. Table 2 details the device count and die area used to provide the 68040 factory test functions.

Ad-Hoc Test Mode:

Section	No. of Devices
Bus Controller	100
Instruction Memory Controller	500
Data Memory Controller	500
Test Interface	260

Scan Test Mode:

Section	No. of Devices
Bus Controller	460
Instruction Memory Controller	410
Data Memory Controller	1,100
Floating Point Unit	6,600
Instruction Unit	8,100
Test Interface	530

Totals:

Total devices Required by Test.....	18,560
Percentage of total number of devices	1.55%
Percentage of total Die Area*	3.15%

* - Includes Global Test Signal Routing

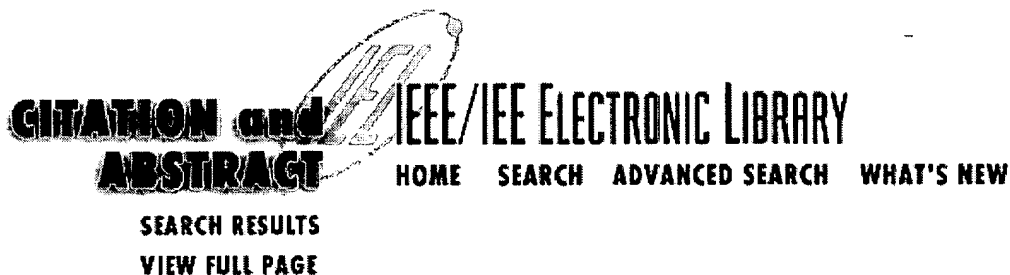
Table 2 - Test Logic Statistics

Acknowledgments

The authors would like to express thanks to Dr. Tom Williams of IBM for his inspiration and to Dr. Ray Mercer of the University of Texas for suggestions and guidance. The authors would also like to thank fellow Motorolans Bill Bruce, Grady Giles and John Salick for their suggestions and direction. We would also like to thank the 68040 design group because without their dedication to the test philosophy the testability of the 68040 would not have been as successful.

References

- [1] Robin W. Edenfield et. al. "The 68040 Processor Part 1", IEEE Micro, February, 1990, pp.66-78
- [2] Robin W. Edenfield et. al. "The 68040 Processor Part 2", to be published in IEEE Micro, June, 1990 issue
- [3] Tom Glover, Motorola, Inc. "Mixed Mode ATPG Under Input Constraints", to be presented at the 1990 International Test Conference, Washington D.C.
- [4] Donnie Anderson et. al. "The 68040 32-bit Monolithic Processor", to be published in the IEEE Journal of Solid State Circuits October, 1990 issue
- [5] J. R. Kuban and J. E. Salick, "Testing Approaches in the MC68020", VLSI Design, Nov. 1984, pp. 22-30.
- [6] L. Basto, P. Harrod, and W. Bruce, "Testing the MC68030 Caches", Proc. IEEE International Test Conference, 1987, PP. 826-833.
- [7] van de Goor, A.J., "State of the Art Memory Testing from Chips to Boards", 1988 International Test Conference, Tutorial Notes, 1988.
- [8] E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability", 1977 Design Automation Conference, New Orleans
- [9] E. J. McCluskey, "A Survey of Design for Testability Scan Techniques", VLSI Design. December, 1984, pp.38-61
- [10] IEEE Standard 1149.1 - 1990. IEEE Service Center Attn.: Sales, 445 Hoes Lane, Piscataway, NJ 08855-1331.
- [11] MC68040 Designer's Handbook, MC68040DH/AD, Chapter 8, Motorola, 1990.



Using boundary scan test to test random access memory clusters

- *Muris, M.; Biewenga, A.*

Philips Electron. Design & Tools, Eindhoven, Netherlands

This Paper Appears in :
Test Conference, 1993. Proceedings., International

on Pages: 174 - 179

This Conference was Held : 17-21 Oct. 1993
 1993 ISBN: 0-7803-1430-1

IEEE Catalog Number: 93CH3356-3

Total Pages: xii+1065

References Cited: 5

Accession Number: 4985252

Abstract:

Interconnects inside clusters of random access memories on printed circuit boards can be tested using boundary scan test. This paper makes recommendations regarding the design of clusters, containing either static or dynamic memory devices, allowing for a complete structural test of the cluster. No additional access to the internal interconnects of the cluster is required. Examples of static and dynamic cluster designs are presented illustrating the recommendations. <

Subject Terms:

boundary scan testing; random-access storage; printed circuit testing; printed circuit accessories; static memory devices; DRAM; SRAM; static cluster designs; boundary scan test; random access memory clusters; printed circuit boards; dynamic memory devices; structural test; internal interconnects; dynamic cluster designs

Help

Copyright © 1997 IEEE -- All rights reserved

USING BOUNDARY SCAN TEST TO TEST RANDOM ACCESS MEMORY CLUSTERS

Math Muris

Alex Biewenga

Philips Electronic Design & Tools
PO Box 80000, 5600 JA Eindhoven
The Netherlands

ABSTRACT

Interconnects inside clusters of random access memories on printed circuit boards can be tested using Boundary Scan Test. This paper makes recommendations regarding the design of clusters, containing either static or dynamic memory devices, allowing for a complete structural test of the cluster. No additional access to the internal interconnects of the cluster is required. Examples of static and dynamic cluster designs are presented illustrating the recommendations.

INTRODUCTION

Complex printed circuit boards are currently designed with components having Boundary Scan Test (BST)[1]. Implementation of BST in Application Specific Integrated Circuits (ASICs) is possible for almost any technology from almost any vendor. Major microprocessor manufacturers are offering new processors with BST already implemented[3,4,5]. Using these microprocessors and ASICs, a high penetration of BST on complex printed circuit boards is possible. However, a problem is still presented by memory devices, like Static Random Access Memory (SRAM), Dynamic Random Access Memory (DRAM), Video Random Access Memory (VRAM), Electrically Erasable and Programmable Read Only Memory (EEPROM), Read Only Memory (ROM) and First-In First-Out memory (FIFO). Although frequently occurring on complex printed circuit boards, it is not likely that all these components will become available in a BST version. The main reasons for this are the variety of different memory types and the pin compatibility problems introduced by adding the 4 or 5 pins of the Test Access Port (TAP) of the BST architecture.

Although memory devices can be very large (i.e. several megabytes in a chip) their basic functionality is easy: using an addressing mechanism, data can be stored in, and/or read from, any point in the address space. This simple basic functionality makes it feasible to develop efficient algorithms for testing the assembly of memory clusters on a printed circuit board[2]. Similar to the interconnect test between BST elements, the test of the memory clusters is targeted at detecting faults in the interconnection of the memory devices, such as open pins and bridges between two nets. During the process of validating these algorithms with respect to their fault detection and fault diagnostic capabilities we discovered a number of recommendations, which should be followed for the design of memory clusters.

In the remainder of this paper, we will present the set of recommendations to be followed for the design of Random Access Memory clusters (Static or Dynamic) on printed circuit board, to be tested from the cluster boundaries by means of BST. We will start with describing the different memory clusters containing different types of memory devices and other supporting devices. Then we will present the recommendations for memory cluster design. Using two examples, one based on SRAMs, the other based on DRAMS, we will indicate how random access memory clusters following the rules can be tested efficiently with Boundary Scan Test.

RANDOM ACCESS MEMORY CLUSTERS

Random Access Memory devices come in two basic flavours:

- static,
- dynamic.

The Static Random Access Memory (SRAM) devices are available in a large range of sizes (Kbytes to Mbytes) and in a wide range of different versions (unidirectional data pins versus bidirectional data pins, dual-ported, several select mechanisms using one or more chip select lines or output enables, optional reset pins, built in parity bits, etc.). A static device does not impose maximum duration constraints on its read or write cycles. Once data has been written into the device, it will be retained there as long as the power supply is available.

The Dynamic Random Access Memory (DRAM) devices are also available in a large range of sizes (Kbytes to Mbytes) but are standardized on one version with an access protocol based on Row Address Select (RAS) and Column Address Select (CAS) pins. A dynamic device does impose constraints on the maximum active time of the RAS and CAS pulse width during read and write cycles. Typically these maximum times vary between 10.000 ns and 100.000 ns. Furthermore, each address has to be accessed within a fixed time period to recharge the capacitor holding the data value. This maximum time, usually referred to as 'refresh time', is in the order of 1 ms to 100 ms.

We define a random access memory cluster as a collection of non-BST devices, containing either static or dynamic Random Access Memories. Besides the non-BST memory devices the memory cluster can contain any of the following non-BST devices:

- transparent/clocked data bus transceivers
- address latches
- chip select decoders

Our assumption is that the inputs and outputs of a memory cluster are accessible through BST cells of the surrounding BST components. If different types of memories (e.g. DRAM and SRAM) are connected to the same data bus or address bus, then the different memory types will be tested as separate clusters. A memory cluster may contain several banks of memory devices. All banks are connected to the same address and data bus. At each moment in time not more than one bank is allowed to drive on the data bus to avoid bus contention (= memory read). However, usually it will be possible to load data into all the banks at the same time (= memory write). Each memory bank may contain several memory devices in parallel to extend the

width of the data bus. For example, 4 devices of 32k x 8 bit may be used to create one memory cluster of 32k x 32 bits

DESIGN RECOMMENDATIONS FOR MEMORY CLUSTERS

1 *Provide sense/drive access to all the nets of the data bus.*

Access to the data bus is required during read and write cycles. This may be provided by a single BST cell on one data line, able to drive and sense a bidirectional pin, or by one cell for driving data on the bus through a tri-state output pin and one or more cells for sensing the value on the data bus. Note, that fault identification resolution is improved, by also reading back the data from the bus during a write cycle into the memory.

2 *Provide direct control to the other drivers on the data bus.*

It may occur that multiple drivers are present on the data bus, some of which may not be directly controlled by BST. In this case, means should be provided to force these drivers from the data bus under direct control of BST. E.g. a microprocessor without BST can be forced off the data bus by driving its reset or hold input pin directly by a BST cell.

3 *The use of non-BST data bus transceivers is allowed.*

If transceivers are placed on the data bus, in between the memories and the driving/sensing BST cell, they will be implicitly tested when they are used during the read and write cycles from and to the memory. Note however, that the diagnostic resolution is improved when the data bus transceivers are equipped with BST.

4 *Multiple banks are allowed.*

In a memory cluster, often multiple memory devices can drive onto the same data bus. Provided that the bank selection is directly controlled from the BST register, all banks can be tested separately, with 100% fault coverage.

5 *Provide pull up resistors on all nets of the data bus.*

It is recommended to provide pull up (or pull down) resistors on all nets of the data bus. This will increase the number of potential detecting patterns for interconnect faults affecting the control lines

which can enable the outputs of the memories on the data bus. It will not increase the detection percentage (that can always be 100% for stuck-at's, given appropriate test patterns for a cluster with multiple banks), but it will increase the diagnostic resolution of the patterns. If transceivers are used in the data path then the pull up resistors should be placed on the data bus between the memory chips and the transceivers.

6 *Provide drive access to the address bus.*

Drive access to the address bus is required during read and write cycles. Note, that the address bus must be driven in a glitch-free way, because changes on address lines while the 'write enable' control line is active, will result in a write cycle on the 'old' address.

7 *Provide drive access to all control lines.*

Usually there are a number of control lines which have to be driven to perform read or write cycles on a memory device. Access to all of these should be provided for test purposes.

8 *The use of bank select logic is allowed.*

For decoding the high order address bits into chip select lines for the different memory banks, often simple decoders are used (2-to-4 or 3-to-8). Any manufacturing defects on the interconnects to these components will be detected by suitable test patterns for the available memory banks. Note however, that the diagnostic resolution will be increased by replacing the decoder with a decoder containing a BST circuit.

9 *Provide the means to tristate the RAS (Row Address Strobe) and the CAS (Column Address Strobe) lines of dynamic memories.*

Due to the timing constraints of dynamic memories, it is usually not feasible to control the RAS and CAS lines directly by a BST register cell. Due to the fact that all test patterns have to be shifted in through the BST register, meeting the maximum RAS and CAS pulswidth constraints (e.g. < 10.000 ns) would require excessive high test clock frequencies (e.g. > 25 MHz). This problem may be solved by using an external tester pod for generating the RAS/CAS protocol. Connection of this tester pod is only possible if the regular RAS/CAS driving pins can be disabled directly through the BST register.

EXAMPLES

The following two paragraphs give examples of two memory clusters we examined. The first is an example of a Static memory cluster, the second is an example of a Dynamic memory cluster.

SRAM

This board has four 32K x 8 bit standard Static RAM memory ICs. Each IC has fifteen address lines, eight data lines, a write enable, an output enable and a chip enable. Further more there is a 2 x 4 address decoder and there is an 8 bits address latch. Figure 1 shows the memory cluster on the board.

We had drive and sense access to all the nets marked by the dots (i.e. C1, G1, A0-A7, D0-D7, etc.). The access was via the surrounding Boundary Scan components (not drawn).

The test patterns were generated using the algorithms presented in "Memory Interconnection Test at Board Level" [2]. This resulted in 1609 test patterns. These test patterns were applied to the cluster through a BST register of the surrounding components, which had a length of 128 bits. With the tester running at 5 MHz the approximate test execution time was 43 ms.

A fault simulation of the test patterns predicted a 100% fault detection of all single stuck-at faults. To verify the prediction of the fault simulator we applied all possible faults to the cluster (this included not only the stuck-at faults but also all bridging combinations (bridges between 2 nets)). The result was that we did indeed detect all the applied faults.

The fault simulator also produced a fault dictionary based on the single stuck-at model. The diagnosis performed with this fault dictionary correctly diagnosed all the stuck-at faults. The diagnosis, as was to be expected, had a decreased diagnostic resolution when bridging faults were applied to the cluster. For certain bridges the diagnosis tool only found one half of a bridge (for example: if address line A5 was bridged with data line D3 the diagnosis tool would only identify A5 as stuck at 1). For other bridges it found both halves of the bridge, identified as stuck-at one (or zero as the case may be). This diagnostic resolution was considered sufficient for our needs in the repair environment for this board. Looking up the fault in the fault dictionary

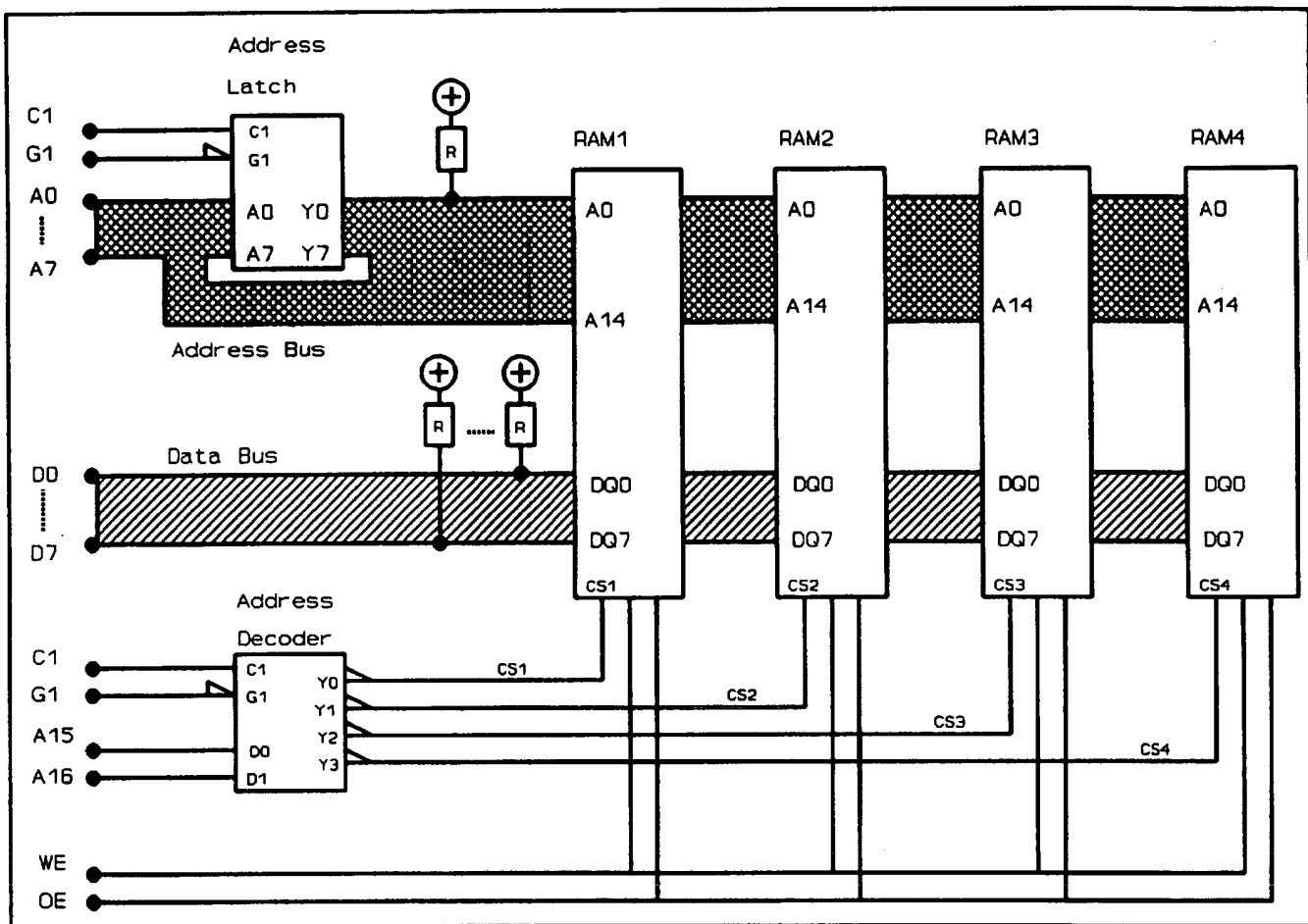


Figure 1. An example of a SRAM board.

took on average 10 seconds run time on a state of the art Personal Computer.

DRAM

Our second example is a board with 8 DRAM chips (1M x 4 bit). The memory cluster consists of 2 banks, each bank with 4 DRAM chips (bank A consists of DRAM1-4 and bank B consists of DRAM5-8, see figure 2). The banks can be selected by means of the RAS and CAS lines (RAS0, CAS0, RAS1 and CAS1).

There is 1 common address bus (10 bits wide) which goes to all the DRAM chips. There is a common write enable (WE) and a common output enable (G).

Each bank is split into a low byte and a high byte. DRAM1 and DRAM2 form the low byte of bank A, DRAM3 and DRAM4 form the high byte of bank A. We can read and write data in words or bytes by

enabling just one or both of the transceivers.

The dynamic requirements which we had to meet for this DRAM cluster were twofold:

- 1 RAS and CAS pulswidth less than 100,000 ns
- 2 All 1024 row addresses refreshed within 16 ms

From these two requirements it followed that we had to generate the RAS and CAS timing with a tester pod, external to the board.

We had access to all the nets on the left of the board (access is marked by the dots). We were able to disable the RAS and CAS signals produced by the memory controller on the board and generate these by an external tester pod. Our RAS and CAS signals were produced by a KAS (Keep Alive Scan module) which is controlled by the Boundary Scan Tester. The KAS is capable of generating 4 RAS and 4 CAS signals, in

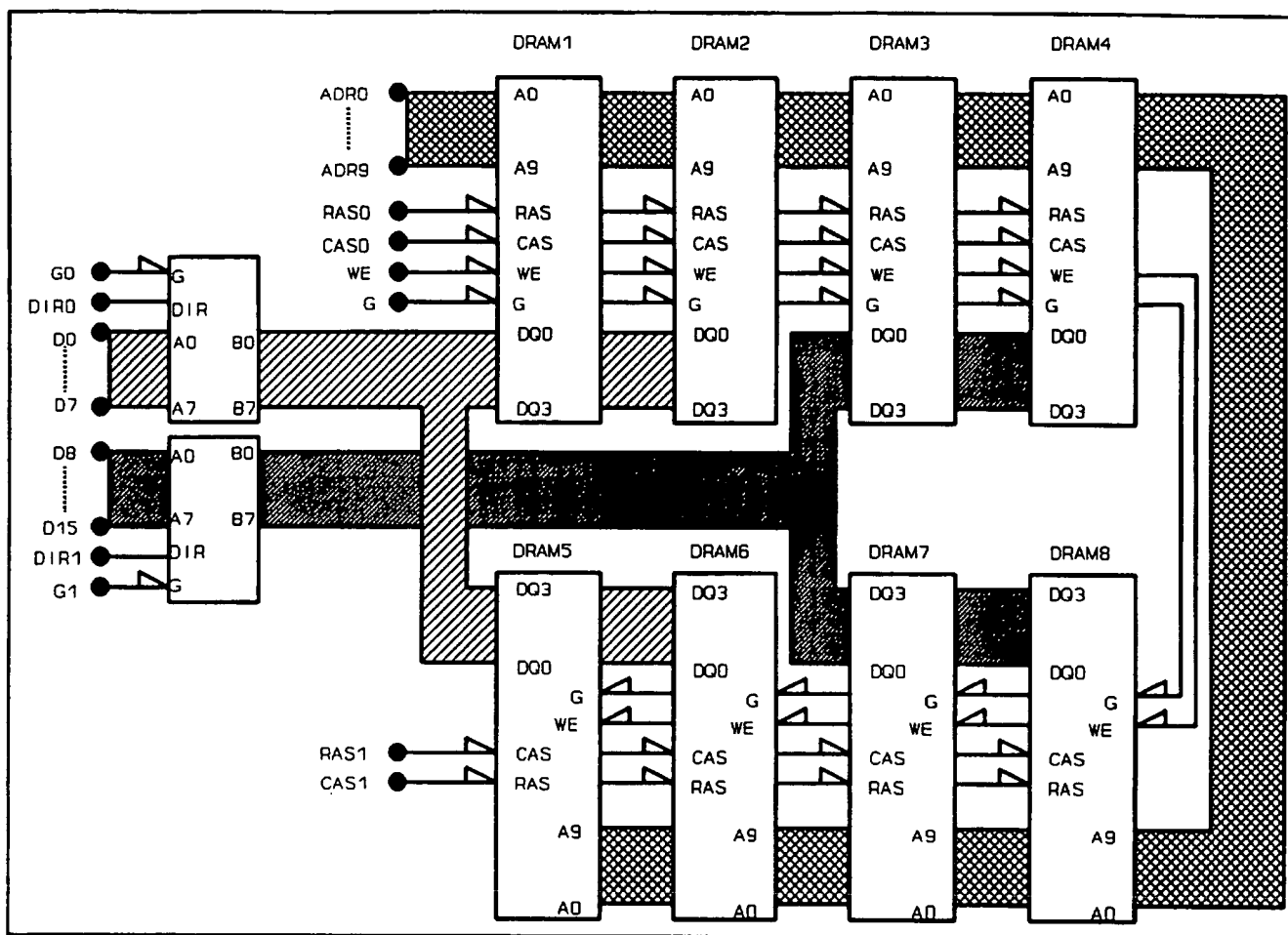


Figure 2. An example of a DRAM board.

synchronisation with the TCK signal of the TAP. The KAS can generate the RAS-before-CAS read/write protocol, the CAS-before-RAS refresh protocol and the RAS only refresh protocol.

The test patterns were generated using the same method as used for the SRAM cluster. This resulted in 160 test patterns. These test patterns were applied to the cluster through a BST register of the surrounding components, which had a length of 128 bits. With the tester running at 5 MHz the approximate test execution time was 4ms.

A fault simulation of the test patterns again predicted a 100% fault detection of all single stuck-at faults. We also verified this prediction by inserting all possible faults to the cluster (this included not only the stuck-at faults but also all bridging combinations (bridges between 2 nets)). The result was that we did indeed detect all the inserted faults.

The fault simulator also produced a fault dictionary based on the single stuck-at model for the DRAM cluster. The diagnosis performed with this fault dictionary correctly diagnosed all the stuck-at faults. The diagnosis, as with the SRAM diagnosis, had a decreased diagnostic resolution when bridging faults were applied to the cluster. For certain bridges the diagnosis tool only found one half of a bridge (see SRAM example).

CONCLUSION

In this paper, we have shown that Random Access Memory Clusters on a board can be tested from the surrounding Boundary Scan components without additional access to the internals of the cluster. A set of recommendations are provided for the design of random access memory cluster, covering both static as dynamic memory clusters. Patterns can be generated with a fault coverage of 100% for all possible stuck-at faults on pins

and interconnects and with a fault coverage of 100% of all possible bridging faults between any two nets of the cluster. Also the presence of non-BST devices like data bus transceivers and bank select decoders is allowed, without affecting the fault coverage. Fault diagnostic resolution however is increased when BST devices are used as data bus transceivers or bank select decoders. We furthermore verified that fault diagnosis of memory clusters, based on fault dictionaries created for single stuck-at fault models, also were suitable for locating bridging faults in the memory cluster.

ACKNOWLEDGEMENTS

The authors wish to thank their colleagues who helped them to verify the discussions put forward in this paper.

REFERENCES

- [1] IEEE Standard 1149.1-1990, "Standard Test Access Port and Boundary-Scan Architecture", IEEE Standards Board, New York, 1990.
- [2] De Jong, F., De Lind van Wijngaarden A.J., "Memory Interconnection Test at Board Level", Proceedings of the International Test Conf., 1992, pp 328-337.
- [3] Bruce, W., Gallup, M., Gilles, G., Munns, T., "Implementing 1149.1 on CMOS microprocessors", Proceedings of the International Test Conf., 1991, pp. 879-886.
- [4] "Intel Microprocessors", 1992, Volume I, chapter 2, "Intel 486TM Microprocessor".
- [5] "Intel Multimedia and Supercomputing processors", 1992, chapter 2, "i860TM Microprocessor family".

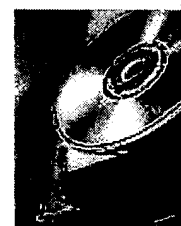
**CITATION and
ABSTRACT**

IEEE/IEE ELECTRONIC LIBRARY

HOME SEARCH ADVANCED SEARCH WHAT'S NEW

SEARCH RESULTS

VIEW FULL PAGE



Unified scan design with scannable memory arrays

- Yano, S.

1st Comput. Oper. Unit, NEC Corp., Tokyo, Japan

This Paper Appears in :
Test Symposium, 1995., Proceedings of the Fourth Asian

on Pages: 153 - 159

This Conference was Held : 23-24 Nov. 1995
1995 ISBN: 0-8186-7129-7

IEEE Catalog Number: 95TB8084

Total Pages: xix+393

References Cited: 14

Accession Number: 5291468

Abstract:

Scan design has been popular as a design-for-testability technique. A memory array, however, has been considered non-scannable. This paper describes unified scan design that makes a memory array scannable and allows mixing of memory arrays and ordinary flip-flops in a single scan path. Based on a rule that considers ordinary flip-flops as a memory array with one word, the existing CAD system can generate the test pattern automatically without making any distinction between flip-flops and memory arrays. A long scan path involving a number of memory arrays can be split into multiple scan paths to reduce scan operation time.

Subject Terms:

design for testability; logic CAD; integrated memory circuits; automatic testing; fault diagnosis; flip-flops; arrays; shift registers; unified scan design; scannable memory arrays; design-for-testability; flip-flops; single scan path; scan operation time; scannable register file

Help

Copyright © 1997 IEEE -- All rights reserved

Unified Scan Design with Scannable Memory Arrays

Seiken Yano

NEC Corporation, 1st Computers Operations Unit
1-10, Nisshin-chou, Fuchu-shi, Tokyo 183, Japan

Abstract

Scan design has been popular as a design-for-testability technique. A memory array, however, has been considered non-scannable. This paper describes unified scan design that makes a memory array scannable and allows mixing of memory arrays and ordinary flip-flops in a single scan path. Based on a rule that considers ordinary flip-flops as a memory array with one word, the existing CAD system can generate the test pattern automatically without making any distinction between flip-flops and memory arrays. A long scan path involving a number of memory arrays can be split into multiple scan paths to reduce scan operation time.

1. Introduction

Scan design and its variations have been widespread as a design-for-testability technique [1]-[6]. In scan design, flip-flops are connected serially in order to be used as pseudo-I/O terminals in testing. By applying scan design, we can transform a sequential circuit into a combinational one. Automatic test pattern generation for a combinational circuit is well understood and is less complex than functional test for a sequential circuit. It is faster, requires less memory, and provides higher stuck-at fault coverage than other techniques.

Recently, major processor designs have adopted scan design for testing their random logic [7]-[10]. Embedded memory arrays such as register files, however, have been considered non-scannable. To apply scan design to a memory array, all bits of memory cells should be included in a scan path. However, cell array configuration prevents a memory array from providing scan path ability.

Therefore, other techniques such as BIST (built-in self test) [11]-[13], ad-hoc direct memory access, and functional test have been used to test embedded memory arrays. BIST is effective in detecting a fault, but it cannot diagnose the fault. Ad-hoc direct memory access and functional test require greater design efforts compared to the structured scan path approach. Consequently, test

design occupies a large portion of the LSI development time.

To overcome this drawback, unified scan design will be presented in this paper. Unified scan design allows us to mix memory arrays and ordinary flip-flops in a single scan path. The application of unified scan design to an LSI having a number of memory arrays would result in a very long scan chain. We could divide the long scan chain into multiple scan paths to reduce scan operation time. Finally, based on a unified rule that considers flip-flops as a memory array with one word, test pattern can be generated by automatic test pattern generator for mixed environment consisting of random logic and memory arrays.

Since 1968, when NEC introduced the first scan design [1], various forms of scan elements have been proposed. Among them, the most widely applied scan design depends on the use of scan elements that comprise edge triggered flip-flops and two-input multiplexers. In the design, a scan mode control signal is employed to switch between the scan and normal operation modes, and the system clock is used to perform scan operation. This standard environment is assumed in this paper.

2. Scannable memory array

2.1 Configuration of a scannable register file

A register file is one example of a memory array, and is used as a data storage for arithmetic and logic operations in a processor. First, a register file with one write and one read ports will be discussed to demonstrate scan path ability. The register file assumes that it stores write-data synchronized to the leading edge of the clock signal, and data in a memory location designated by read-address is available at data output asynchronously to clock signal.

To provide scan path ability to a register file, we have added extra support hardware: a scan-address counter, a write-data multiplexer, and write- and read-address multiplexers, as shown in Figure 1. The Scan-Address

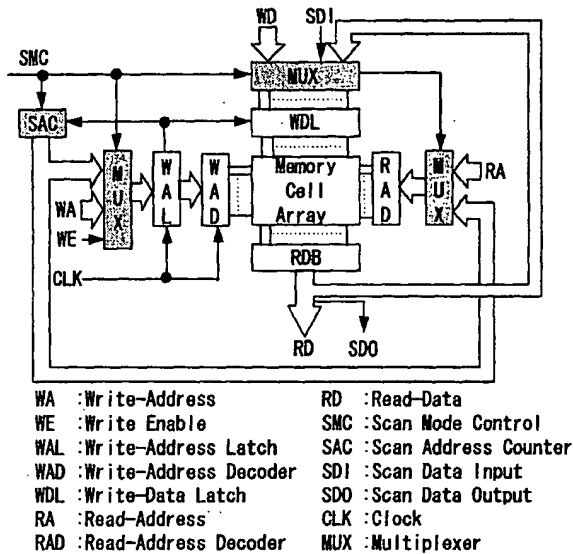


Figure 1. Register file with scan path

Counter (SAC) increments synchronized to clock in scan mode (SMC=1), and its value at the initiation of the scan operation is always zero. The address multiplexers select the SAC output in scan mode. When the scan mode is true, the write-data multiplexer selects the read-data which is shifted right by one bit position and added scan-in data at the most left position.

As for a multiple port register file with more than one read and/or write ports, we can apply the approach by choosing one write and one read ports for scan path ability. We insert a write-data multiplexer, and write- and read-address multiplexers to the selected ports. Then, small circuits are added to the non-selected write ports to inhibit write operation in scan mode. No additional circuits are needed for non-selected read ports.

An example of a scannable register file with two write

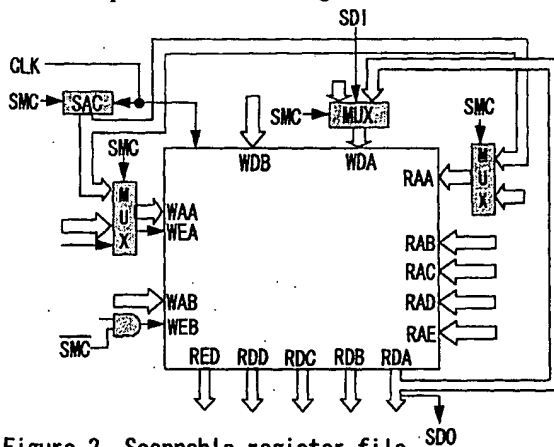


Figure 2. Scannable register file with 2 write and 5 read ports

and five read ports is illustrated in Figure 2. Here, Write Port A and Read Port A are selected for scan operation and Write-Enable to Write Port B is inhibited in the scan mode. The resulting scan operation is same as that of a register file with one read and one write ports.

2.2 Scan operation of a scannable register file

As a simple example, assume that a register file has four words of four bits each. Figure 3(a) shows the SAC and memory cell status at the initiation of the scan operation. The contents of the register file are shown as A, B, C, ..., P and the bits shifted in through Scan Data Input (SDI) will be shown as a, b, c, ..., p, where these characters represent logical "1" or logical "0". We define bit position as "Bit 0", "Bit 1", etc. from right to left and word position as "Word 0", "Word 1", etc. from top to bottom.

Initially, the SAC points to "Word 0" and scan-in data is "a". The data read out from "Word 0" is "M-I-E-A", the scan-out data from "Bit 0" position is "A", and the write-data ready to be written into "Word 0" is "a-M-I-E". The SAC increments and points to "Word 1", "Word 2", and "Word 3" according to clock advances, as shown in Figure 3(b), 3(c) and 3(d), respectively. After four clocks have advanced, the register file is changed into the state shown in Figure 3(e). Scan-in data have replaced four initial bits in "Bit 3" position from "Word 0" to "Word 3" in temporal order of scan-in. The initial contents of memory cells have been shifted right by one bit position and the initial data at "Bit 0" position have been shifted out sequentially from "Word 0" to "Word 3". The SAC points to "Word 0" again.

After supplying 16 clocks to the register file, initial contents of memory cells have been all shifted out, and have been replaced by scan-in data, as shown in Figure 3(f). Figure 4 shows the scan-out and scan-in data from right to left in temporal order. Both data are aligned in sequential order of bit and word position in the memory array. The scan mechanism described in this section can be extended to a register file of arbitrary size and number of read and write ports.

3. Serial connection of memory elements

3.1 Serial connection of register files

Serial connection of register files concerning scan path presents a problem in case of register files with different numbers of words and/or bits. If these register files have the same number of words and differ only in bit width, we can consider them as a single register file with wider bit width. An issue exists in the case of serial connection of register files with different word size.

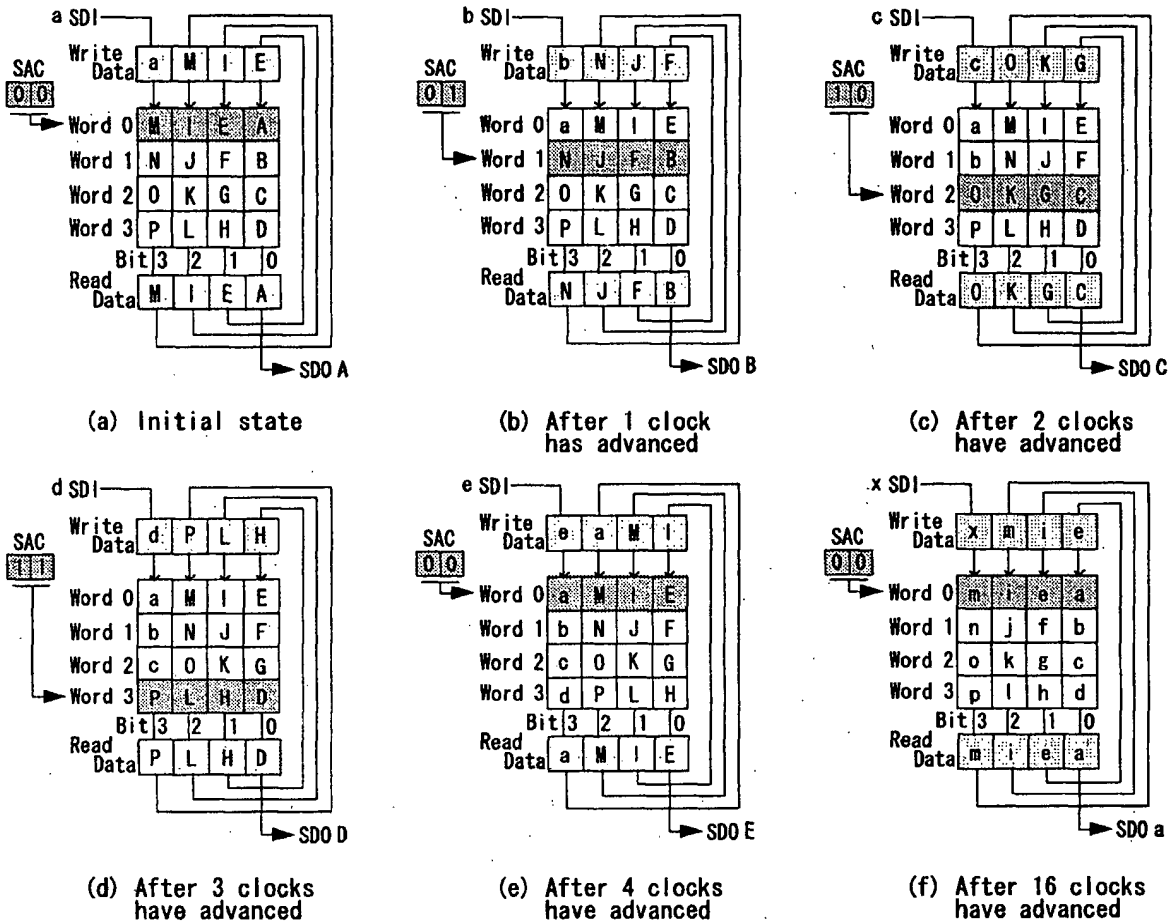


Figure 3. Register file scan operation (Status of memory cells and SAC are indicated)

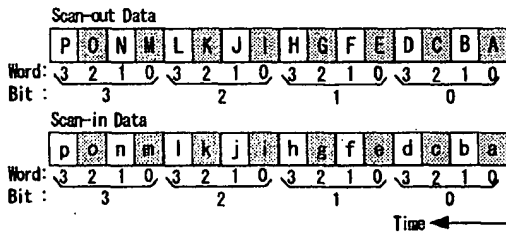


Figure 4. Scan data of scannable register file

As an example, assume that Register File A with four words by four bits is connected to Register File B with three words by two bits, shown in Figure 5. This figure illustrates the status at the initiation of the scan operation, and SAC in the register files points to "Word 0". The connected register files have 22 bits totally (16 bits of Register File A and 6 bits of Register File B), and twenty-

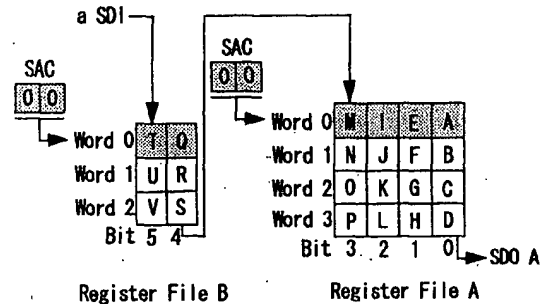


Figure 5. Serial connection of register files (at the initiation of scan operation)

two clocks are enough to scan out all bits in the array. The temporal scan-out data is aligned in desirable order: ascending sequential order of bit and word position.

However, final state of the array, shown in Figure 6,

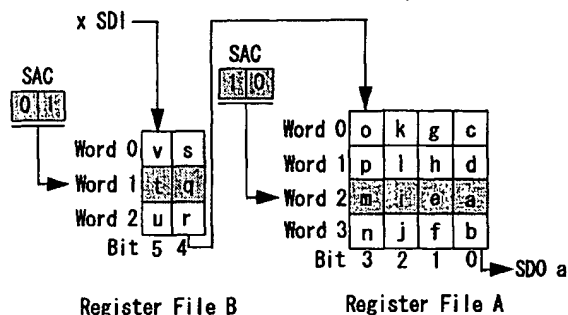


Figure 6. Serial connection of register files (after 22 clocks have advanced)

has a problem. The data from SDI is not stored in desirable sequential order: For example, "Word 1" in "Bit 0" holds data "d" that is shifted in after data "a" in "Word 2". Coincidence of word position order and scan-in temporal order is inevitable for CAD system to generate test pattern independent of the connection state. In order to get coincidence of word position order and scan-in temporal order, we should supply enough clocks so that SACs point to "Word 0" at the end of scan operation. SACs of Register File A and B, however, point to "Word 2" and "Word 1", respectively. In this case, to ensure the SACs point to "Word 0" at the end of scan operation, we should supply two more clocks: totally 24 clocks. The number twenty four corresponds to an integer multiple of both four (words of Register File A) and three (words of Register File B).

Accordingly, two dummy bits should be added to scan data. Scan-in data has two dummy bits followed by 22 data bits. Scan-out data should be also 24 bits, because scan-out operation could be accomplished simultaneously with scan-in operation of next test vector to reduce LSI tester time. As a result, the last two bits of expected scan-out data are "don't cares". Complete scan-out and scan-in data have desirable sequential order in bit and word position, as shown in Figure 7.

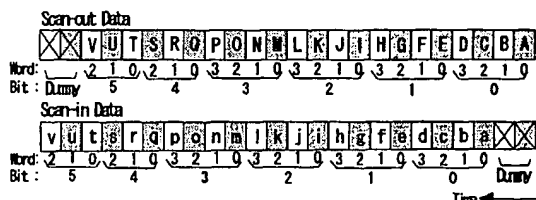


Figure 7. Scan data with dummy bits

Consequently, we can implement scan path ability to serial connection of register files with different words by satisfying the integer multiple requirement: the number of total scan bits must be an integer multiple of the number of words of any register file included in the scan path.

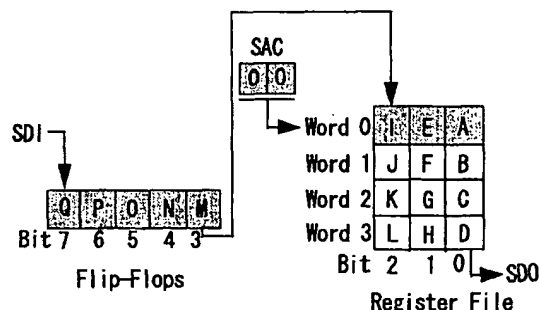


Figure 8. Serial connection of a register file and ordinary flip-flops

3.2 Serial connection of register files and flip-flops

To expand scan design beyond random logic, it is necessary to mix ordinary flip-flops and memory arrays in a single scan path. In figure 8, we illustrate one example of serial connection of a register file and flip-flops: five bits of flip-flops connected to a register file with four words by three bits. We can consider this configuration as a special case of serial connection of register files with different words: the chain of five flip-flops is equivalent to a register file with one word by five bits.

Therefore, by considering flip-flops as a register file with one word, we can generate test pattern by the same manner as described in the previous section. This example has 17 bits totally (5 bits of flip-flops and 12 bits of the register file), we should adjust scan data to have 20 bits: an integer multiple of four (words of the register file) and one (word of flip-flops). The test pattern generated based on the rule mentioned above is shown in Figure 9. After scan operation, the status of the example has desirable order of bit and word.

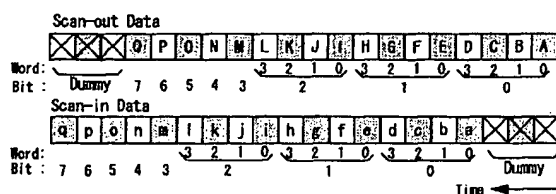


Figure 9. Scan data with dummy bits

4. Multiple scan paths

When a number of register files are included in a scan path, the scan chain may become very long making the scan operation slow. Generally speaking, a processor may integrate several thousands of memory elements excluding a large memory array such as a cache memory, which would be tested by ad-hoc direct access method or

BIST. A long scan path could be divided into multiple scan paths in order to reduce scan operation time. Today, we have LSI testers that can handle up to 32 scan channels and up to 512 million words of scan data, where one word consists of three bits: scan-in, scan-out and care/mask bit for scan-out. Therefore, we could divide a long scan path into up to 32 shorter paths. The LSI tester allows sharing of scan-in and scan-out signals with normal operation signals.

To demonstrate multiple scan paths, assume that we have two scan paths: Scan Path 1 has Register File A with four words by three bits and Register File B with three words by two bits, and Scan Path 2 has five bits flip-flops and Register File C with four words by four bits. Figure 10 shows the status at the beginning of scan operation. The contents of memory elements are shown as A1, B1, C1, ..., R1 for Scan Path 1, and A2, B2, C2, ..., U2 for Scan Path 2. The bits shifted in through SDIs will be shown as a1, b1, c1, ..., r1 for Scan Path 1, and a2, b2, c2, ..., u2 for Scan Path 2.

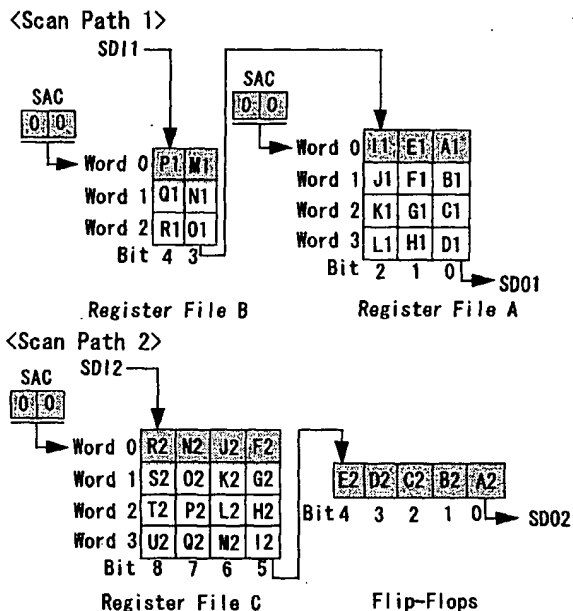


Figure 10. Multi-scan path

In our environment, we employ the scan mode control to switch between the scan and normal operation modes, and use the system clock to perform scan operation. These signals are distributed to all scan elements, independent of scan path configuration. This means the same number of system clocks should be supplied to all scan elements during the scan operation. Therefore, each scan path must have the same number of scan bits. Furthermore, all SACs of register files should point to "Word 0" at the end of scan operation to get desirable

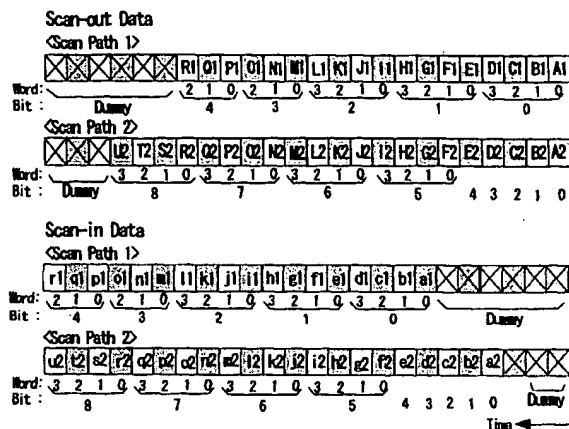


Figure 11. Scan data for multi-scan

sequential order of bit and word. As a result, the number of scan bits for a scan path should satisfy an integer multiple condition not only for the register files in the scan path, but also for the register files in the other scan chains.

Scan Path 1 and 2 have 18 bits and 21 bits, respectively. Therefore, the number of scan data should be more than 21 and an integer multiple of 4 (words of Register File A and C), 3 (register File B) and 1 (word of flip-flops). We add six dummy bits for Scan Path 1, and three dummy bits for Scan Path 2. Test pattern of 24 bits is generated to satisfy the above mentioned condition, as shown in Figure 11.

In this example, all flip-flops are included in Scan Path 2. Usually we could distribute flip-flops among many scan paths to satisfy the integer multiple condition. By distributing flip-flops, we could realize the best scan path configuration that would minimize the number of dummy bits.

5. Unified scan test

From the descriptions in previous sections, we can derive test pattern generation rules for unified scan design as follows.

- (1) Consider ordinary flip-flops as a register file with one word.
- (2) The number of scan bits of a scan path must be an integer multiple of the number of words of any register file included in all scan paths. If necessary, add dummy bits to satisfy the integer multiple condition.
- (3) When adding dummy bits, place the bits at the front of scan-in data and discard the extra bits at the end of scan-out data.

The Scan Address Counter, added for testing purpose only, is not included in the scan path, and is invisible to logic designers. The unified scan design makes it possible to detect all stack-at faults in memory cells and peripheral circuits, as well as in random logic. This can eliminate time consuming design efforts that would be required to test memory arrays and their boundary with random logic. Furthermore, the new technique makes it easy to diagnose a memory array. For example, one bit stack-at fault of a memory cell causes a cyclic error in scan-out data, and the error can be distinguished from an error in flip-flops.

One of the main features of the unified scan design consists in its higher access rate to memory arrays. In scan mode, scan data goes through all memory arrays, and those memory arrays execute a read/write operation at every clock advance. Assume that scan data is one thousand bits long, we have one thousand read/write operations to memory arrays followed by one normal operation. This means that more than 99.9% of tester time is spent for memory array testing. Therefore, we can state that the unified scan test is a memory test which features test data with higher stack-at fault coverage for peripheral combinational circuits including random logic. Conceptual block diagram of an LSI with unified scan design is shown in Figure 12.

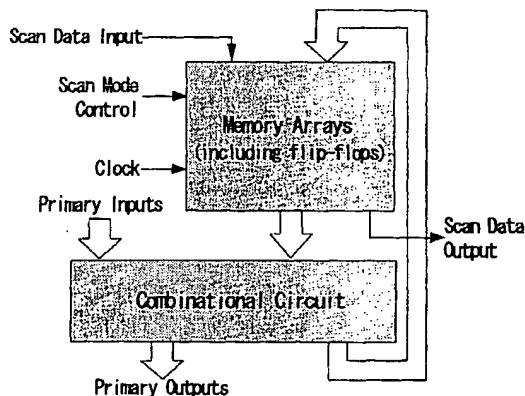


Figure 12. An LSI with unified scan design

6. Application

The unified scan design has been applied to a gate array LSI with embedded scannable register files. It features bipolar LCML (Low-energy Current Mode Logic), 458 I/O pads (including 321 signal pads) for TAB assembly, and internal area with 16 macro blocks arranged in a four by four matrix: eight gate array macros and eight register file macros [14]. The LSI microphotograph is shown in Figure 13, and main features are listed in Table 1.

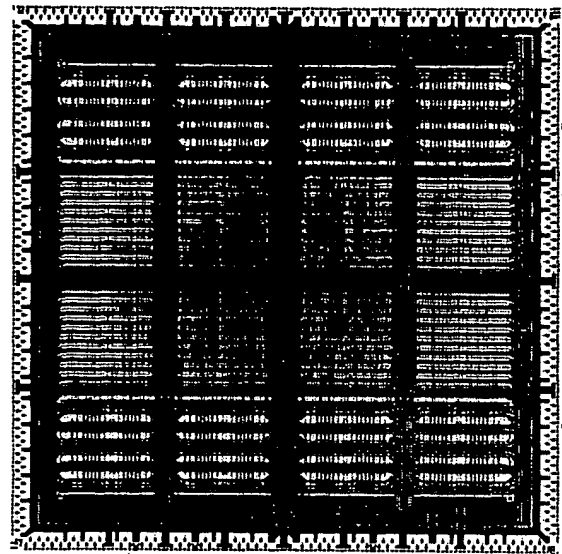


Figure 13. Microphotograph of the gate array with scannable register

- ◆ Basic gate — LCML (Low-energy CML)
- ◆ Gate count — 21,000 (Max)
- ◆ I/O pads — 458 (Signal 321)
- ◆ Die size — 14×14 mm
- ◆ Power supply — 4.5V, -3.0V, -2.0V

Table 1. Gate array with scannable register files

The gate array macro contains 160 cells including 42 bits of flip-flop cells and 118 LCML gate cells. The register file macro has one write and one read ports, and its configuration is 16 words by 5 bits by 4 banks: totally 320 bits. Each bank has its own write-enable signal, so we can realize such a register file with 16 words by 20 bits, 32 words by 10 bits, or 64 words by 5 bits. However, the multiple number for scan data adjustment is 16 regardless of these configuration. If any macro is not used in option design, we can replace the macro by a dummy one that has no transistor connection to reduce power dissipation. Eventually, the dummy macro is not included in the scan path.

The LSI integrates totally 83,000 transistors, including 5,200 transistors for scan path. Therefore, the overhead ratio for scan design is about 6.7% in transistor count. As for power dissipation, the overhead is mainly in register file macros and estimated to occupy about 3.5% of total power dissipation. The support hardware also affects flip-flop set-up time, and the degradation due to scan design is approximately 1.2% of a cycle time.

We have developed 68 metal options for supercomputer SX-3, mainframe ACOS 3800, and ACOS 3900. Design data of these LSIs shows that average cell usage of 87.5% results in 19,000 equivalent gates, and 36.6W of power dissipation, as listed in Table 2. The test pattern shows average fault coverage of 99.2%. Maximum scan path length is 2,896 bits: 336 bits of flip-flops and 2,560 bits of register files. Scan bits of most LSIs, however, are less than 2,896 bits depending on their function, and are adjusted by adding dummy bits so that the number of scan bits equals to an integer multiple of sixteen.

◆ Cell usage	-- 87.5%
◆ Equivalent gates	-- 19,000
◆ Signal pads	-- 282
◆ Power (TYP.)	-- 36.6W
◆ Fault coverage	-- 99.2%

Table 2. Design data of 68 metal options
(mean value)

7. Conclusion

In order to overcome problems concerning memory array testing, unified scan design has been presented. This new approach is supported by adding extra hardware to a memory array: a scan address counter, a write-data multiplexer, and read- and write-address multiplexers. With the support hardware, we can make a memory array scannable and mix memory arrays and ordinary flip-flops in a single scan path. The hardware also makes a multi-port register file scannable in the same way as a register file with one read and one write ports.

When a number of memory arrays are included in a scan chain, the scan path can become very long. We could divide the chain into multiple scan paths to reduce scan operation time. Ordinary flip-flops could be distributed among scan paths to adjust the number of scan bits so as to reduce the number of dummy bits.

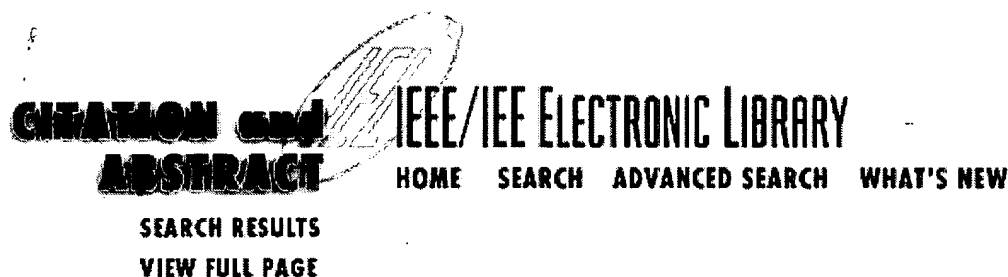
The test pattern can be generated based on a rule that considers ordinary flip-flops as a register file with one word. The number of scan bits in a scan path should equal to an integer multiple of the number of words of any register file included in the scan paths. If necessary, we could add dummy bits to satisfy the integer multiple requirement. The dummy bits are placed at the front of scan-in data, and the extra bits at the end of scan-out data are simply discarded.

The new scan design has been successfully applied to LSIs used in a supercomputer and large mainframes. The longest scan chain has 2,896 bits, and test pattern for 68 metal options show average stack-at fault coverage of 99.2%.

The significant aspect of the new scan design is the fact that it offers a unified scan path approach in an environment with random logic and memory arrays in a single chip. As a result, the approach allows ATPG (Automatic Test Pattern Generator) to generate test patterns for random logic and memory arrays. This would eliminate time consuming ad-hoc design efforts that would be required to test memory array and its boundary with random logic.

References

- [1] A. Kobayashi, S. Matsue, and H. Shiba, "Flip-Flop Circuit with FLT Capability," Proc. IECEO Conf., 1968, p.692 (in Japanese).
- [2] S. Funatsu, N. Wakatsuki, and T. Arima, "Test Generation Systems in Japan," Proc. Design Automation Conf., 1975, pp. 114-122.
- [3] E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability," Proc. Design Automation Conf., 1977, pp. 462-468.
- [4] T. W. Williams, and K. P. Parker, "Design for Testability - A Survey," Proc. of the IEEE, Vol. 71, No. 1, pp. 98-112, January 1983.
- [5] S. Funatsu, M. Kawai, and A. Yamada, "Scan Design at NEC," IEEE Design & Test of Computers. Vol. 6, No. 3, pp. 50-57, June 1989.
- [6] B. Könemann, J. Barlow, P. Chang, R. Gabrielson, C. Goertz, B. Keller, K. McCauley, J. Tischer, V. Iyengar, B. Rosen, and T. Williams, "Delay Test: The Next Frontier for LSSD Test Systems," Proc. Int. Test Conf., 1992, pp. 578-587.
- [7] M. G. Gallup, W. Ledbetter Jr., R. McGarity, S. McMahan, K. C. Scheuer, C. G. Speard, and L. Sood, "Testability Features of the 68040," Proc. Int. Test Conf., 1990, pp. 749-757.
- [8] D. D. Josephson, D. J. Dixon, and J. Arnold, "Test Features of the HP7100LC Processor," Proc. Int. Test Conf., 1993, pp. 764-772.
- [9] C. Huter, E. Kofi Vida-Torku, and J. LeBlanc, "Balancing Structured and Ad-hoc Design for Test: Testing of the PowerPC603 Microprocessor," Proc. Int. Test Conf., 1994, pp. 76-83.
- [10] A. L. Crouch, M. Pressly, and J. Circello, "Testability Features of the MC68060 Microprocessor," Proc. Int. Test Conf., 1994, pp. 60-69.
- [11] B. Nadeau-Dostie, A. Silburt, and V.K. Agarwal, "Serial Interfacing for Embedded-Memory Testing," Design & Test of Computers, Vol. 7, No. 2, pp. 52-63, April 1990.
- [12] K. Kinoshita, and K. K. Saluja, "Built-in Testing of Memory Using an One-chip Compact Testing Scheme," IEEE Trans. on Computers, Vol. 10, pp. 862-870, Oct. 1986.
- [13] M. Franklin, and K. K. Saluja, "Built-in Self-Testing of Random-Access Memories," Computers, Vol. 23, No. 10, pp. 45-56, Oct. 1990.
- [14] M. Katagiri, T. Dasai, S. Iriki, Y. Ohyama, K. Hasegawa, S. Nakahara, and H. Matsumoto, "High-Speed LSIs for NEC Supercomputer SX-3," NEC Res. & Develop., Vol. 33, No. 2, pp. 161-169, April, 1992.



Effects of simultaneous switching noise on the tapered buffer design

- Vemuru, S.R.

Dept. of Electr. Eng., City Univ. of New York, NY, USA

This Paper Appears in :
Very Large Scale Integration (VLSI) Systems, IEEE Transactions on

on Pages: 290 - 300

Sept. 1997

Vol. 5

Issue: 3

ISSN: 1063-8210

References Cited: 35

CODEN: IEVSE9

Accession Number: 5689553

Abstract:

Complementary metal-oxide-semiconductor (CMOS) output buffers, comprised of a series of tapered inverters, are used to drive large off-chip capacitances. The ratio of the size of transistors between two consecutive stages is the buffer taper factor. With higher frequency of operation and simultaneous switching of the output drivers, the parasitic inductance present at the pin-pad-package interface results in significant switching noise on the power lines. A comprehensive analysis and estimate of simultaneous switching noise (SSN) including the velocity saturation effects seen in the submicron transistors during the switching of output drivers is presented. The effect of SSN on the overall buffer propagation delay and transition time is discussed. The presence of SSN results in an increase in the optimum taper factor between inverter stages for a given capacitive load. Beyond a critical value, the output transition time of a tapered buffer increases with reducing taper factor due to SSN. SSN can be reduced by skewing the switching of output buffers, SPICE simulation results show that skewing buffer switching with additional inverter stages reduces SSN and increases buffer propagation delay.

Subject Terms:

integrated circuit noise; simultaneous switching noise; tapered buffer design; CMOS

output buffers; off-chip capacitances; parasitic inductance; pin-pad-package interface; velocity saturation effects; submicron transistors; buffer propagation delay; inverter stages; output transition time; SPICE simulation

Help

Copyright © 1997 IEEE -- All rights reserved

Effects of Simultaneous Switching Noise on the Tapered Buffer Design

Srinivasa R. Vemuru, *Member, IEEE*

Abstract—Complementary metal-oxide-semiconductor (CMOS) output buffers, comprised of a series of tapered inverters, are used to drive large off-chip capacitances. The ratio of the size of transistors between two consecutive stages is the buffer taper factor. With higher frequency of operation and simultaneous switching of the output drivers, the parasitic inductance present at the pin-pad-package interface results in significant switching noise on the power lines. A comprehensive analysis and estimate of simultaneous switching noise (SSN) including the velocity saturation effects seen in the submicron transistors during the switching of output drivers is presented. The effect of SSN on the overall buffer propagation delay and transition time is discussed. The presence of SSN results in an increase in the optimum taper factor between inverter stages for a given capacitive load. Beyond a critical value, the output transition time of a tapered buffer increases with reducing taper factor due to SSN. SSN can be reduced by skewing the switching of output buffers. SPICE simulation results show that skewing buffer switching with additional inverter stages reduces SSN and increases buffer propagation delay.

I. INTRODUCTION

THE CURRENT trend in high-performance integrated systems design is toward achieving higher speeds. Buffer circuits are typically used in the presence of large capacitive loads such as the off-chip capacitance, large on-chip capacitances seen in clock-lines, large fanout circuits, and long busses, to reduce signal delays and to increase the speed of operation. The focus of this article is the CMOS off-chip driving buffers used to reduce delays due to large capacitive loads of chip-to-chip communication lines. A typical CMOS buffer consists of a series of tapered inverters, with each inverter driving another large inverter until an inverter large enough to drive the load capacitance within a reasonable amount of time is obtained [1], [2]. With increasing frequency of operation of modern digital circuits, the effects of the parasitic inductances are becoming very significant. A typical output stage of an integrated circuit (including the final inverter stage of the output buffer) is shown in Fig. 1(a) along with the parasitic inductances, resistances and capacitances present at the pad-pin and pin-package boundaries. Due to the large slew rates of currents flowing through these inductances during output switching transitions, the supply voltages seen by the output driver vary resulting in a switching noise. This noise in the supply voltages is further enhanced by simultaneous

switching of output drivers resulting in simultaneous switching noise (SSN). SSN on power lines can propagate to other circuits that share the same power bus and can result in glitches in remote receiver circuits. Many approaches are present in the literature to analyze and optimize the tapered buffer designs with respect to number of performance measures such as the propagation delay, power dissipation, area, and hot-electron reliability. We present the effects of the simultaneous switching noise on the performance measures of tapered buffers. In Section II, we introduce CMOS tapered buffers and the methods used in the analyses and optimization of tapered buffers. In Section III, we present expressions for simultaneous switching noise using α -power law model. The effects of SSN on output transition time, propagation delay, and taper factor of the buffer are discussed in Section IV. Some techniques used to reduce the effects of SSN in buffer design are presented in Section V.

II. CMOS TAPERED BUFFERS

CMOS buffers consist of a series of tapered inverters with each inverter driving a larger inverter as shown in Fig. 2 [1]. In typical buffer designs, the ratio of the transistor sizes between two consecutive inverter stages, the taper factor, F , is constant. Jaeger derived the optimum taper factor (to minimize overall buffer propagation delay) using a simplified RC delay model to represent the driving transistor and driven gate capacitive load [2]. The taper factor and the number of inverter stages are given by the following expressions:

$$F = \frac{W_{i+1}}{W_i}; \quad F_{opt} = e = 2.72 \quad (1)$$

$$n = \ln \frac{C_L}{C_y} \quad (2)$$

where C_L is the capacitive load driven by the buffer and C_y the gate capacitance of a unit size inverter. The inverter load capacitance model was improved to include self-loading capacitance of the driving inverter stage and optimum taper factors for minimum buffer delay are derived [3]–[9]. The optimum taper factor for minimizing buffer propagation delay including the inherent capacitance of the driving inverter stage C_x is obtained from the solution of the following expression [6]:

$$F[\ln(F) - 1] = \frac{C_x}{C_y} \quad (3)$$

Recently, the capacitive model was further improved by including the constant component to inherent capacitance, C_x

Manuscript received January 21, 1997. This work was supported in part by the City University of New York PSC/CUNY Research Awards 665334 and 666358 and by the National Science Foundation under Grant CDA-9114481.

The author is with the Department of Electrical Engineering, City College and the Graduate Center of the City University of New York, New York, NY 10031, USA.

Publisher Item Identifier S 1063-8210(97)06355-5.

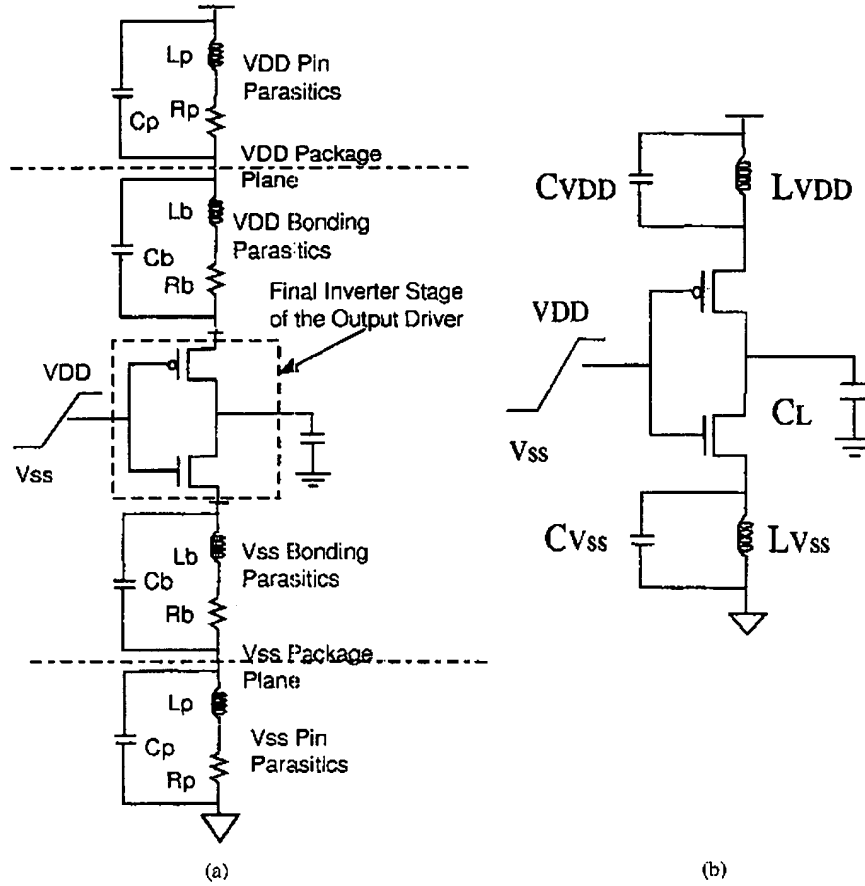


Fig. 1. (a) Pad-pin-package parasitics of the output buffer. (b) Simplified equivalent circuit used in analysis of output buffers.

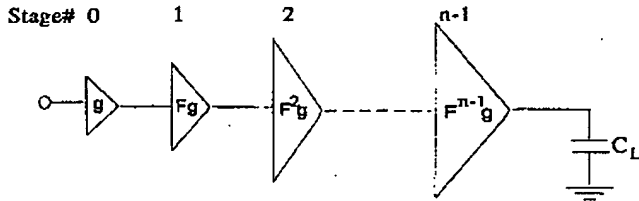


Fig. 2. A fixed taper buffer with a taper factor F .

[9]. Using this model, the optimum buffer taper factor can be obtained from

$$\frac{\ln \frac{C_L}{C_y}}{F(\ln(F))^2} [F'(\ln(F) - 1)C_y - C_x] - \frac{C_x}{(F-1)^2} \left(1 + \frac{1 + \left(\frac{F-1}{F} \right) \ln \left(\frac{C_L}{C_y} \right)}{F \frac{\ln \left(\frac{C_L}{C_y} \right)}{\ln(F)}} \right) = 0. \quad (4)$$

The optimum buffer taper factor is also derived to minimize power dissipation and power-delay product of tapered buffers [5], [10]–[16], buffer area [3], [5], [16]–[20], and hot-electron reliability [16].

III- SSN MODELING

Earlier studies on CMOS tapered buffers have developed models for estimating different performance criteria used in buffer optimization [4], [16], [19]. However, all of these models neglected the effects of parasitic inductances present in the pad-pin-package interface. Output buffers are one of the principal sources of power dissipation in integrated circuits. The power dissipation of input/output (I/O) stages can be as high as 80% of the total power dissipation in the integrated circuit [21], [22]. In typical CMOS designs, multiple power and ground pins are used to supply the large current requirements of I/O pads. These large currents flow through the inductances present at the pad-pin-package interface during the output switching transitions resulting in significant ground bounce. This ground bounce is further enhanced by the simultaneous switching of output buffers present in typical wide-bus architectures of modern microelectronic systems. The presence of SSN reduces the effective gate-source voltage of the output charging/discharging transistor. This decreases the value of current used to estimate propagation delays, transition times, and other performance criteria. Typically the maximum value of SSN is almost independent of the capacitive load of the buffer [23]. However, the SSN is a very strong function of the input transition time, number of switching buffers, and the design of the buffer (taper factor and the transconductance parameter of the driving transistor).

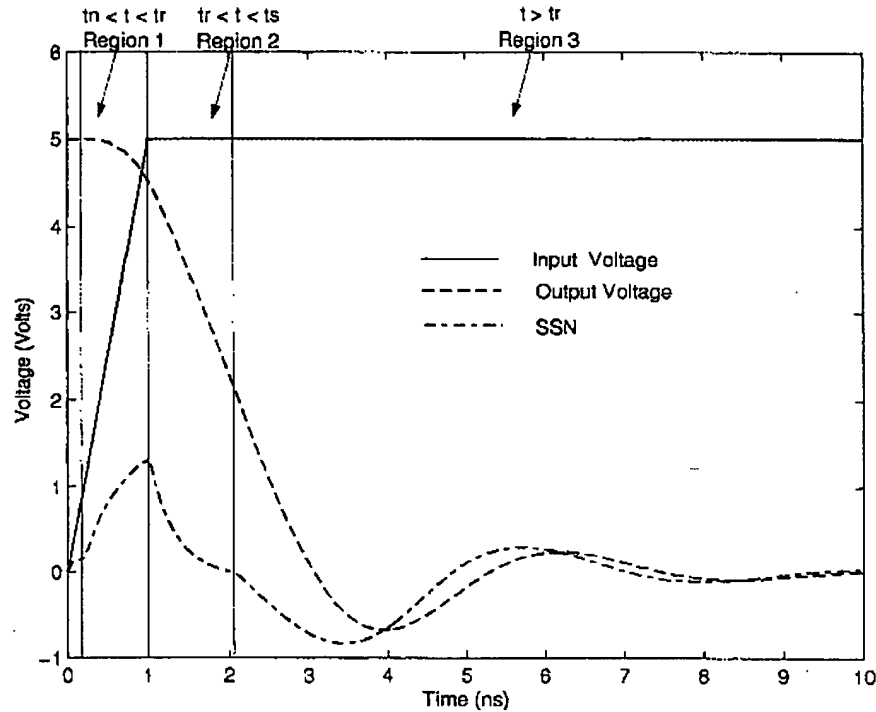


Fig. 3. SSN from SPICE BSIM simulation results of a 0.5- μm CMOS process and classification of regions in SSN analysis.

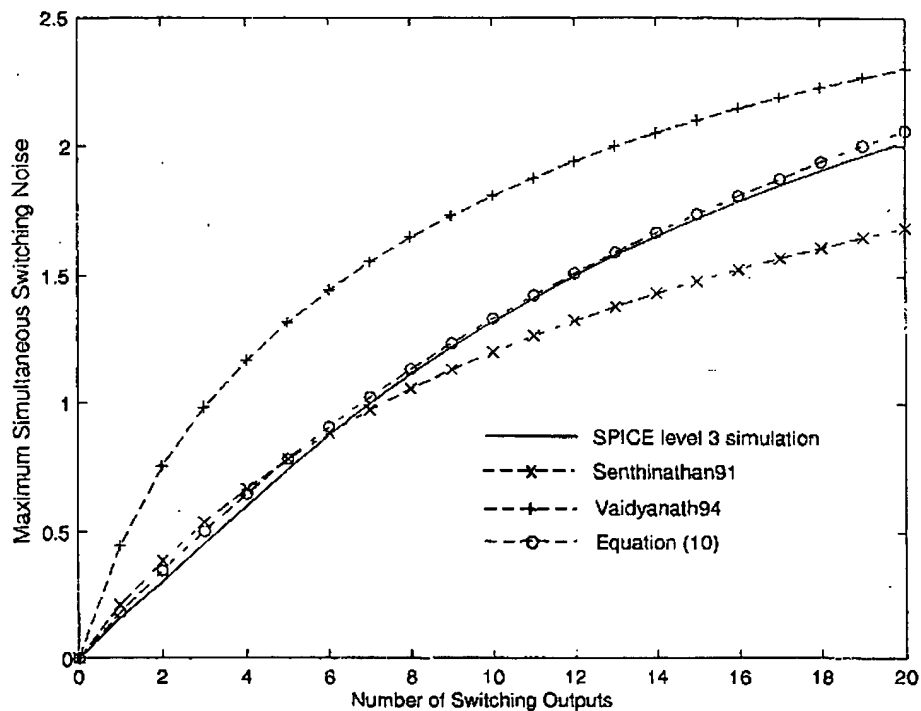


Fig. 4. Comparison of maximum SSN estimates with the SPICE BSIM simulation results.

The analysis of SSN is performed by using a simplified equivalent circuit of the final output driver stage (inverter) as shown in Fig. 1(b). The analysis is presented for the ground bounce in this section. Similar results can be developed for V_{DD} bounce during rising output transitions. In typical fixed taper-buffer designs, the transition times of each of the buffer

stages are comparable. Under these conditions, neglecting the PMOS transistor currents during falling delay analysis result in very small errors [4], [24]. The effect of through current (i.e., the PMOS current during falling output transitions) on switching noise analysis is presented in [23]. With reducing device dimensions and channel lengths, velocity saturation

effects are becoming very significant in short-channel devices. Therefore, the MOSFET does not behave as a square-law device when operating in saturation [25]. The drain current of the MOSFET in the deep submicron region is represented more accurately by a α -power model proposed by Sakurai and Newton [24]. The drain current of a modified version of α -power model is given as [26]

$$I_D = \begin{cases} 0 & (V_{GS} \leq V_{TH}) \text{ cutoff} \\ k_l(V_{GS} - V_{TH})^{\alpha/2} V_{DS} & (V_{DS} < V'_{D0} \text{ linear} \\ k_s(V_{GS} - V_{TH}^\alpha) & (V_{DS} \geq V'_{D0} \text{ sat} \end{cases} \quad (5)$$

$$\begin{aligned} k_l &= \frac{I_{D0}}{V_{D0}(V_{DD} - V_{TH})^{\alpha/2}}; \\ k_s &= \frac{I_{D0}}{(V_{DD} - V_{TH}^\alpha)}; \\ V'_{D0} &= V_{D0} \left(\frac{V_{GS} - V_{TH}}{V_{DD} - V_{TH}} \right)^{\alpha/2} \end{aligned} \quad (6)$$

I_{D0} the drain current when $V_{GS} = V_{DS} = V_{DD}$, is an index of MOSFET current carrying capability and is proportional to its width. V_{D0} is the drain saturation voltage at $V_{GS} = V_{DD}$, V_{TH} is the threshold voltage, and α is the velocity saturation index that accounts for carrier velocity saturation effects. Shichman-Hodges square-law model is a special case of α -power MOS model with $\alpha = 2$. Typical values of α for the deep submicron NMOS transistors are in the range of $1 < \alpha < 1.2$. For a 0.5- μm CMOS process, the extracted α values of the NMOS and PMOS transistors are 1.08 and 1.09, respectively. It is predicted that the value of α approaches unity for channel lengths below 0.5 μm [27].

The input voltage is approximated as a pulse shaped waveform with a specified rise time, t_r , and fall time t_f , and can be expressed as

$$V_{in} = \begin{cases} s_r t & (\text{for rising input}) \\ V_{DD} - s_f t & (\text{for falling input}) \end{cases} \quad (7)$$

where $s_r = V_{DD}/t_r$, and $s_f = V_{DD}/t_f$ are the slopes of the rising and falling input voltages.

A. SSN Estimation

The circuit used to estimate SSN of an inverter is given in Fig. 1(b). For n output buffers switching simultaneously, the switching noise voltage, V_n , is given as

$$V_n = nL_{V_{ss}} \frac{di_o}{dt} \quad (8)$$

where i_o is the current flowing through the driving output transistor. Senthinathan and Prince derived an expression for the SSN by including the effects of the negative feedback of the SSN voltage [28]. They observed that the increase in the switching noise voltage V_n increases the source voltage of the transistor. The increased source voltage acts as a negative feedback to reduce i_o . However the switching current

is approximated as a triangular waveform resulting in an underestimate of SSN. The estimate of SSN was improved by Vaidyanathan *et al.*, who assume that the SSN increases linearly with time [29]. Both of these estimates use the square-law MOSFET model that neglects the velocity-saturation effects in short-channel transistors. A new expression for SSN during rising input voltage transitions using α -power law model for the MOSFET currents was derived [23]. During the falling output voltage transitions the submicron NMOS transistor operates in saturation as long as input voltage signal is rising. The only exception to this would be a very large transistor driving a very small capacitive load; this exception is not valid in the case of off-chip drivers that typically see large capacitive loads. The expression for the SSN is obtained from the solution of (6) using the saturation drain current. The solution of the equation in Region I ($t_n < t < t_r$) is derived in [30] and is given as

$$V_n(t) = s_r n k_{sn} L_{V_{ss}} f [1 - e^{-(t-t_n)/n k_{sn} L_{V_{ss}} f}] \quad \text{for } t_n \leq t \leq t_r \quad (9)$$

where $t_n = V_{TN}/s_r = (V_{TN}/V_{DD})t_r$, and $f = 1.4$. For short-channel MOSFET's, the drain current is a linear function of gate-source voltage and is almost independent of the channel length [27]. Thus, the proposed formula for SSN becomes more accurate for processes with channel lengths below 0.5 μm even with scaled down supply voltages as the velocity-saturation effects are very significant.

The maximum values of SSN is obtained when $t = t_r$ and is given as

$$V_{n,\max} = s_r n k_{sn} L_{V_{ss}} f [1 - e^{-(t_r-t_n)/n k_{sn} L_{V_{ss}} f}]. \quad (10)$$

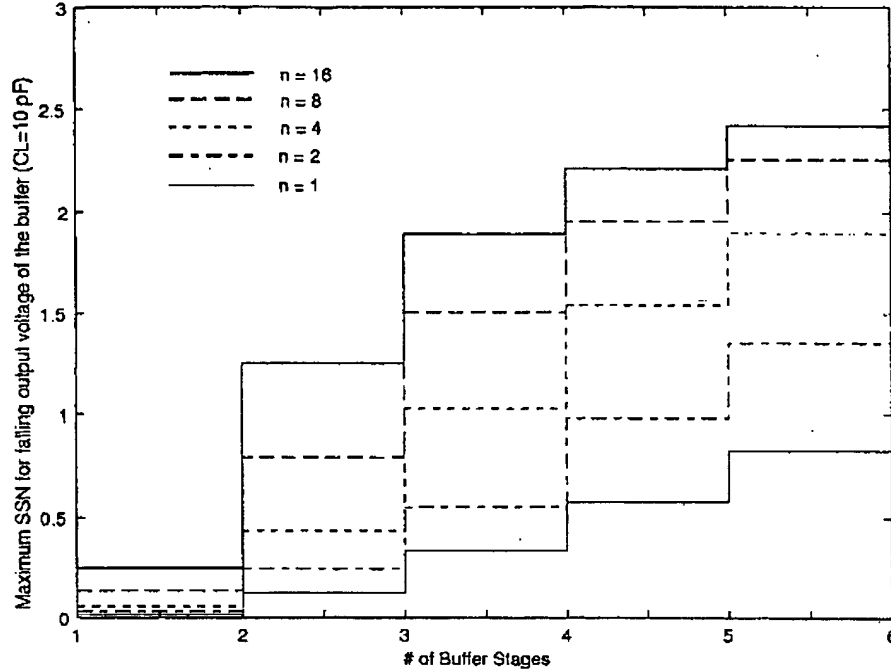
For $t > t_r$, the input voltage remains constant at V_{DD} . This results in a reduced value of SSN as the gate voltage of the driving transistor remains constant. However, the gate-to-source voltage changes due to the reduced switching noise caused by diminished current slew-rate. The differential equation and the solution of the switching noise during this time interval are given as

$$V_n = nL_{V_{ss}} \frac{di}{dt} = nL_{V_{ss}} k_{sn} \frac{d[(V_{DD} - V_{TN} - V_n)]}{dt} \quad (11)$$

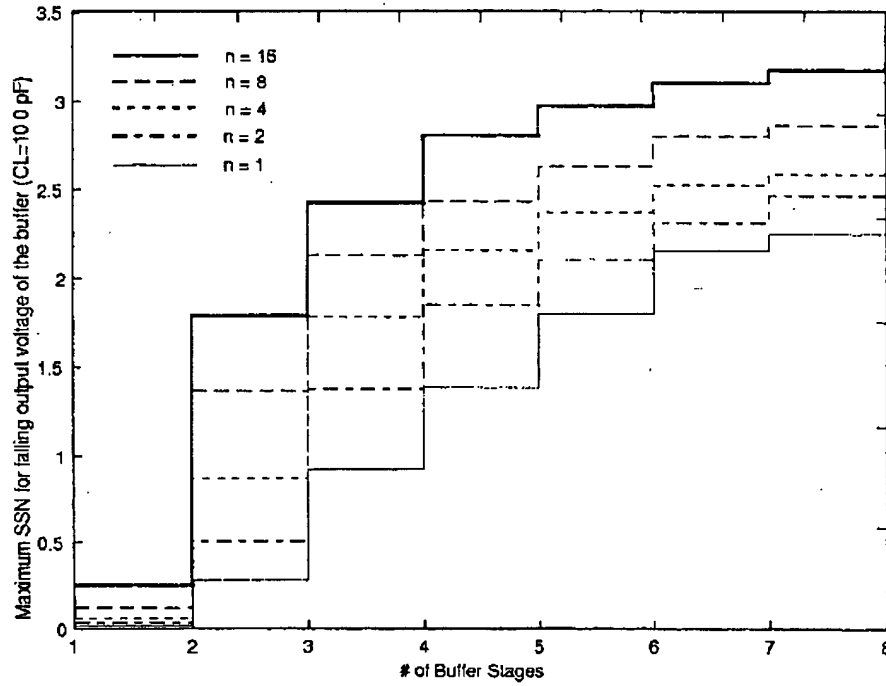
$$V_n(t) = V_{n,\max} e^{-(t-t_r)/nL_{V_{ss}} k_{sn} f} \quad t_r < t \leq t_s. \quad (12)$$

This results in an exponential decrease in the noise voltage as long as the driving transistor is in saturation (Region II— $t_r \leq t \leq t_s$). The driving transistor moves into the linear region (Region III) at the time t_s , and therefore can be replaced by an equivalent resistor, $R = V_{D0}/I_{D0}$. The switching noise in Region III is obtained from the solution of the resulting RLC circuit

$$\frac{d^2 V_n}{dt^2} + \frac{R}{L_{V_{ss}}} \frac{dV_n}{dt} + \frac{1}{L_{V_{ss}} C_L} V_n = 0. \quad (13)$$



(a)



(b)

Fig. 5. Maximum SSN generated in output buffer plotted versus number of buffer stages (taper factor) for different number of simultaneously switching buffers: (a) $C_L = 10$ pF and (b) $C_L = 100$ pF.

For typical packaging systems ($R^2/I_{V_{ss}}^2 < 4/I_{V_{ss}}C_L$). The SSN for $t > t_s$ is given by the following expression:

$$V_n(t) = V_{n,s} e^{-(R/2L)(t-t_s)} \cos \left(\left(\sqrt{\frac{R^2}{L^2} - \frac{4}{LC}} \right) t \right). \quad (14)$$

SSN across a 1-nH inductor due to ten simultaneously switching outputs obtained by SPICE simulations using BSIM

model parameters of a 0.5- μ m CMOS process is given in Fig. 3. The plot is classified into Regions 1, 2, and 3 where it can be clearly seen that (7), (10), and (12) represent the behavior of the SSN adequately.

Fig. 4 compares the maximum SSN estimates using (8) and SSN estimates from [28], [29] with SPICE BSIM MOS model simulation results for a 0.5- μ m process. The new expression gives a more accurate estimate of maximum SSN in the

short-channel transistors with velocity saturation effects. In the absence of velocity-saturation effects, the SSN expressions can become inaccurate as the approximation of α is unity may not be valid [23].

IV. EFFECT OF SSN ON BUFFER PERFORMANCE CRITERIA

From (9), the magnitude of the SSN of an inverter depends on the slew rate of its input voltage. In tapered inverter buffer chain, the output transition time of the driving inverter is the input transition time of the driven inverter. The transition time is a very strong function of the buffer taper factor which can significantly alter the magnitude of switching noise. The presence of the ground bounce or the SSN reduces the gate-source voltage of the driving transistor that is charging/discharging the output capacitance. This results in an increase in the propagation delay and output transition time. SPICE simulation results are presented in this section to study the effects of SSN on the overall buffer propagation delay and buffer output transition time.

In Fig. 5(a) and (b), the maximum SSN of different tapered buffer designs driving capacitive loads of 10 pF and 100 pF, respectively, is plotted. The switching noise is plotted with the number of simultaneously switching output buffers varying from one to sixteen. The maximum SSN is obtained during the switching of the final inverter stage of the buffer. For a given capacitive load, increasing the number of inverter stages reduces the buffer taper factor. Reducing taper factor increases transistor sizes in the final inverter stage. Therefore, the magnitude of the switching noise increases with increasing number of buffer stages as seen in the SPICE simulation results. SSN is also a very strong function of the slew rate of the input voltage. For similarly sized final inverter stages, larger SSN is generated by the buffer driving smaller capacitive load as it will have smaller input transition time. For example, comparing the switching noise generated by a two-stage buffer driving a 100-pF capacitive load with a six-stage buffer driving a 10-pF capacitive load, it can be clearly seen that the latter is generating more switching noise. This is due to its reduced input transition time or increased input voltage slew rate. This increase in SSN is in spite of the final inverter stage of the former buffer being about 20% larger than the latter.

Fig. 6(b) and 6(c) show SPICE simulation plots during the falling and rising output transitions, respectively, of V_{DD} bounce, ground bounce, and voltages at several nodes of a six-stage tapered buffer shown in Fig. 6(a). During the falling output transitions [Fig. 6(b)], the input to the final inverter stage, *out5*, is making a low-to-high transition. The ground bounce reaches its maximum value when *out5* reaches its maximum value during its rising transition. Similar results are seen in Fig. 6(c) for V_{DD} bounce during falling output transition. Some other local peaks in the ground bounce are also indicated by arrows. These peaks occur during the rising transition of the outputs *out1* and *out3*. However, these peaks are very small due to smaller sizes of the switching transistors at these nodes. The nodes making low-to-high transitions, *out1*, *out3*, and *out5*, track the voltage variations of the V_{DD} bus.

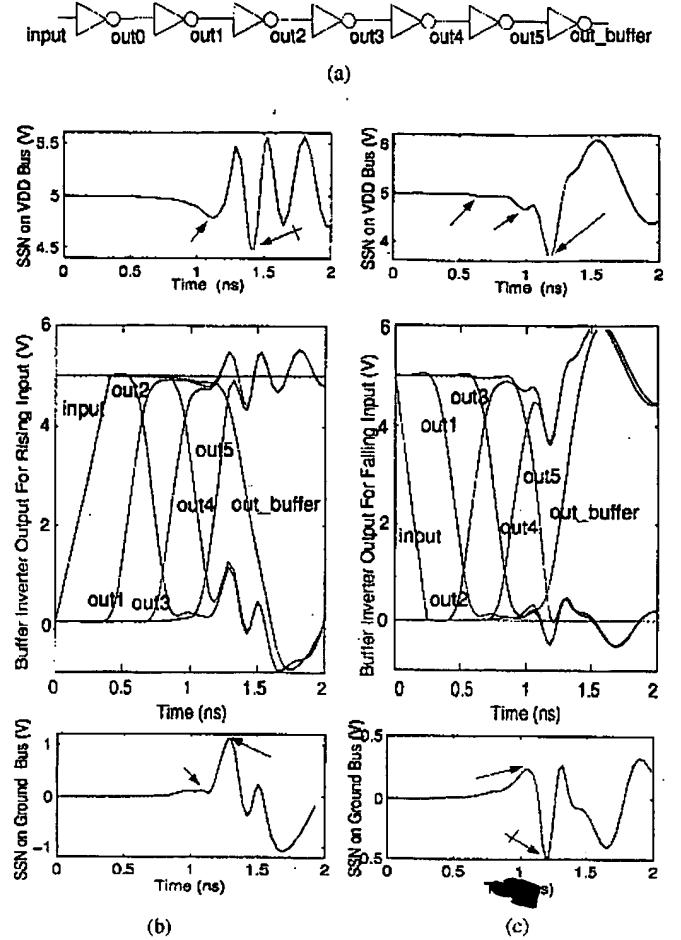


Fig. 6. SPICE simulation results of ground bounce and V_{DD} bounce for a six-stage output buffer driving a C_L of 10 pF; $L = 1$ nH and $n = 1$.

Similarly *out2*, *out4*, and the buffer output, track the changes in the ground bus after making their high-to-low transitions. Therefore, it is very important to reduce the magnitude of SSN as it can cause spurious switching in other circuits that also track the variations in ground/power line when they share the same power bus.

The output transition time of tapered buffers designed to drive capacitive loads of 10 pF and 100 pF, respectively, are plotted in Fig. 7(a) and (b). In the previous studies on buffer transition times, it was observed that the transition time reduces monotonically with increasing number of inverter stages (smaller taper factor) [16]. The sizes of the transistors in the final inverter stage will be larger with reducing taper factor. Therefore they have larger current drive capability that reduces output transition time. However, including parasitic inductances, the current-carrying capacity of the final output stage is affected by the increase in SSN. With reducing taper factors, the transition times of the initial stages of the tapered buffers are reduced. The output transition times of the initial buffer stages are not affected due to relatively smaller magnitudes of SSN during their switching transitions as seen in Fig. 6(b) and (c). Smaller output transition times of the initial buffer inverter stages results in reduced input transition to the later stages. In particular, the transition time

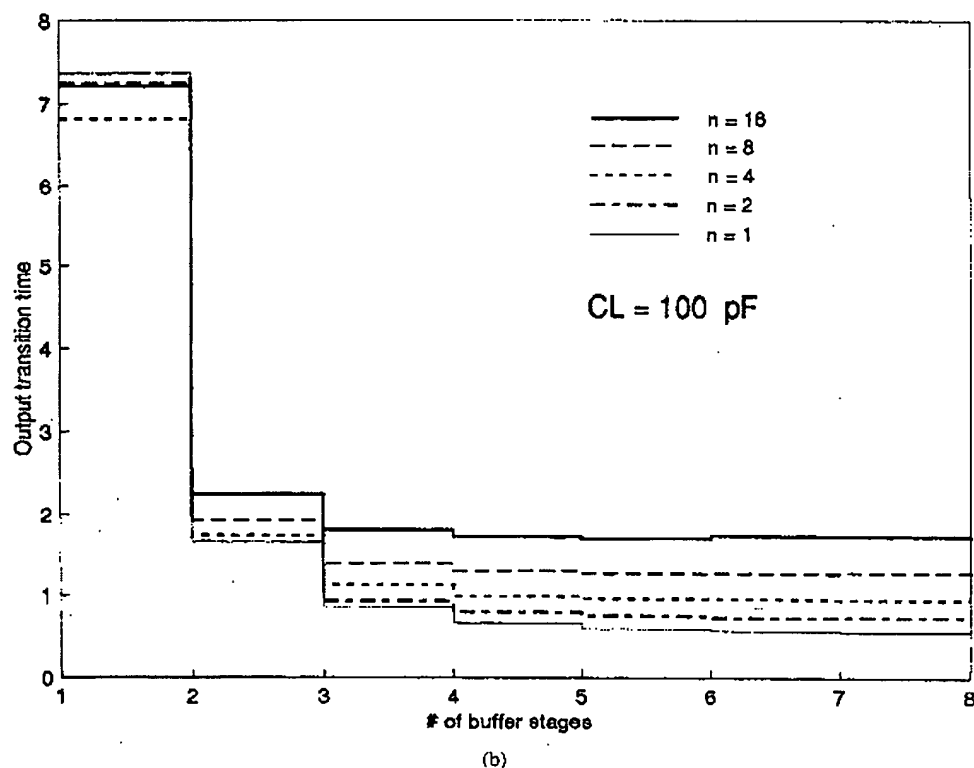
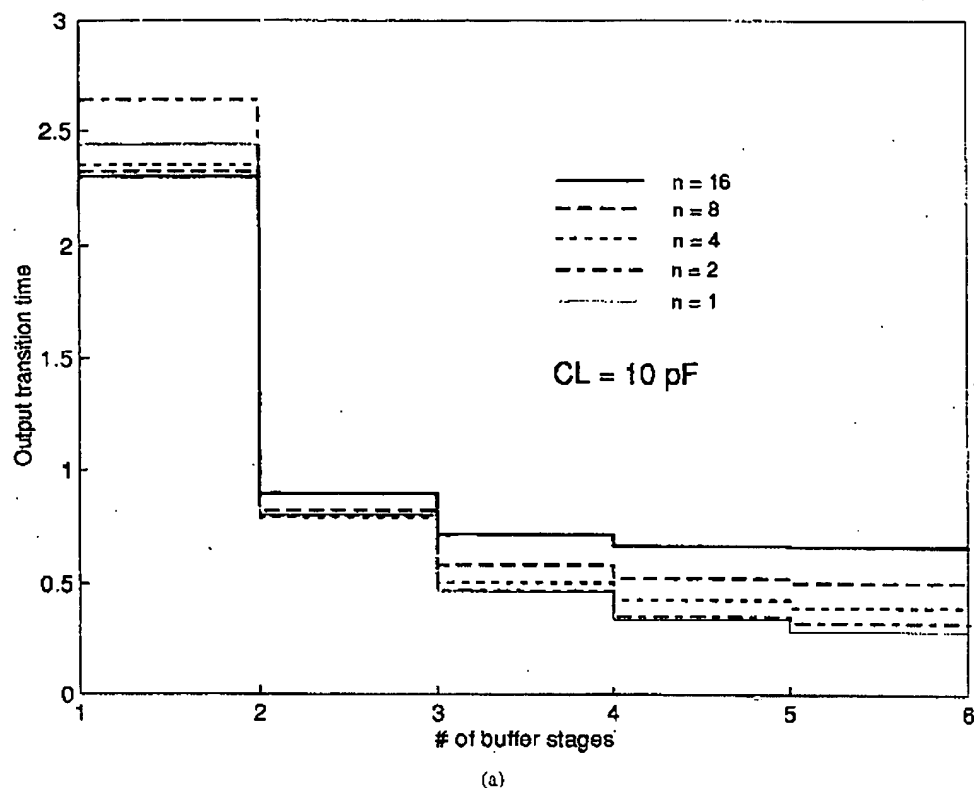


Fig. 7. Output transition time of the buffer plotted versus number of buffer stages (taper factor) for different number of simultaneously switching buffers: (a) $C_L = 10$ pF and (b) $C_L = 100$ pF.

to the final inverter stage is very important as it contributes significantly to SSN. With reduced input transition times, the final inverter stage sees a larger SSN. This reduces its current-driving capacity resulting in an increase in buffer output transition time. For example, the output transition

time of sixteen simultaneously switching seven-stage buffers is greater than sixteen simultaneously switching five-stage buffers designed to drive 100 pF load. This increase in transition time is in spite of the wider transistors used in the final inverter of the seven-stage buffer.

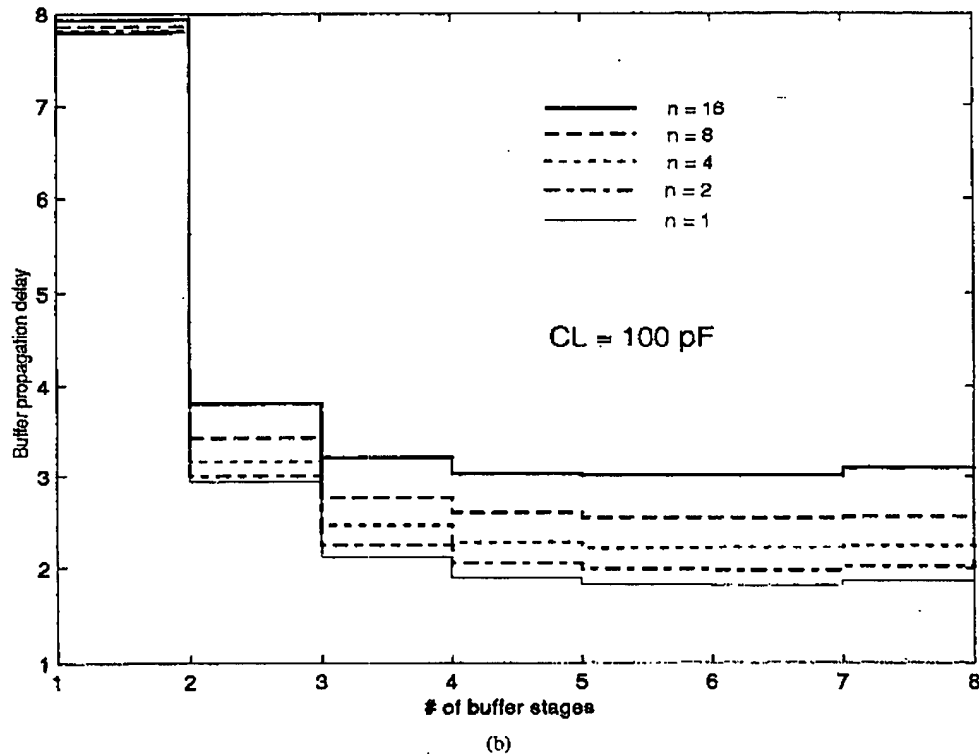
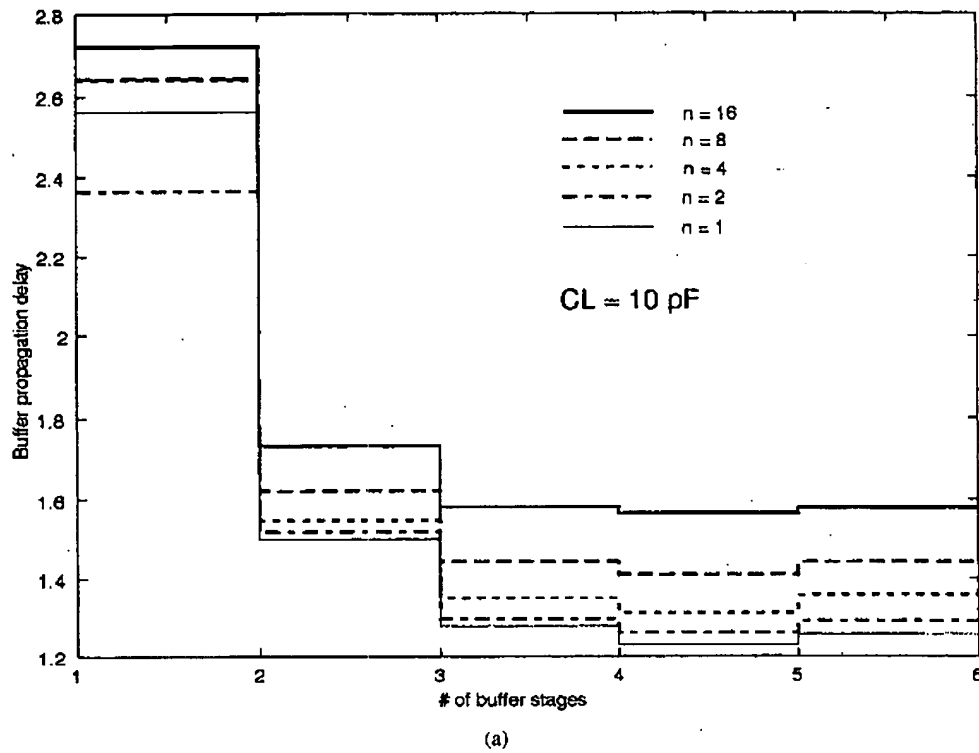


Fig. 8. Buffer propagation delay plotted versus number of buffer stages (taper factor) for different number of simultaneously switching buffers: (a) $C_L = 10 \text{ pF}$ and (b) $C_L = 100 \text{ pF}$.

The effect of supply voltage variations on a logic gate propagation delays have been studied in [31], [32]. In Fig. 8(a) and (b), overall propagation delay is plotted for different buffer designs driving 10 pF and 100 pF , respectively. For 10 pF load, a six-inverter stage buffer gives the minimum propagation

delay if the parasitic inductance effects are neglected [6], [9]. Including inductance-based SSN effects, the optimum number of buffer stages capacitive load was found to be four from SPICE simulations. The resulting taper factor of buffer is 4 in contrast to 3.1 obtained from (4). With reducing taper

TABLE I
EFFECTS OF BUFFER SWITCHING SKEWS ON SSN AND DELAYS

	CL (pF)	# of buffer inverter stages	Case A (No skew)	Case B	Case C
SSN (V)	10	6	2.6013	2.0352	1.6456
	10	4	2.2106	1.6564	1.3520
	100	6	3.1027	3.0436	2.9488
OVERALL BUFFER DELAY (ns)	10	6	1.558	1.775	2.015
	10	4	1.544	1.757	2.099
	100	6	3.078	3.255	3.475

factors, SSN effects on the final inverter stages can increase their propagation delays. This increased propagation delays of the final inverter stages can overcompensate the reductions in propagation delays of initial buffer inverter stages. Therefore reducing number of inverter stages (increasing taper factor) can improve overall buffer propagation delays. Moreover, increased buffer taper factors increases transition time along with the propagation delays of the initial inverter stages. This increased transition times reduce SSN and propagation delay of the final inverter stages. Therefore the number of buffer stages for minimum propagation delays reduce when SSN effects are included in the buffer design. For 100 pF capacitive load, the optimum number of buffer stages for minimum propagation delay without parasitic inductive effects was found to be eight [6], [9]. Including SSN effects of parasitic inductances, the optimum number of buffer stages reduces from eight to six. With increasing number of simultaneously switching buffers SSN effects become even more significant. Therefore the optimum number of buffer stages for minimum propagation delay may become even smaller. For example with 100 pF capacitive load, minimum propagation delay is obtained for five stage buffers for sixteen simultaneously switching buffers. With two simultaneously switching buffers the minimum propagation delay is obtained for six stages for the same load.

V. SSN REDUCTION BY SKEWING BUFFER SWITCHING

In addition to changing buffer propagation delay, SSN on the ground and power buses can cause glitches in quiet output stages. Glitches can sometimes cause spurious switching of quiet buffers and affect the reliable operation of the circuitry. Even in the absence of reliability problems, the power dissipated by the glitches constitutes a significant fraction of total power dissipation in integrated circuits [33]. Therefore it is essential to minimize the magnitude of SSN.

From (9), SSN reduces with reducing number of simultaneously switching final output driver stages. Therefore preventing all output drivers from switching simultaneously reduces the magnitude of SSN can be reduced. A technique to reduce SSN is to skew the switching of output buffers [34]. The skew can be obtained by including additional propagation delays to some buffers. This can be easily achieved in tapered buffer designs by using additional unit-size inverter stages. Three

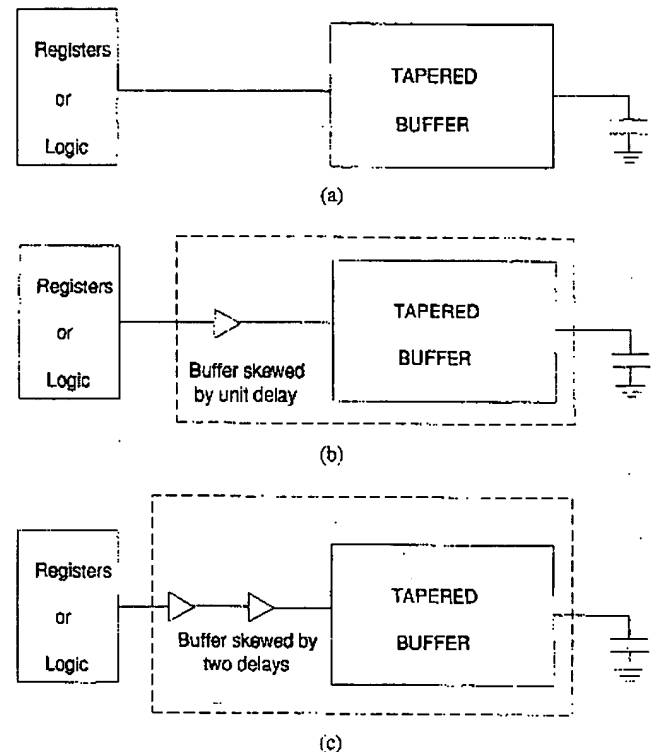


Fig. 9. Modifications made to the tapered buffer to provide skews in simultaneously switching buffers: (a) zero delay, (b) unit delay, and (c) two delays.

different configurations of the tapered buffer and noninverting delay stages (two unit-size inverters in series) to provide zero, one, and two delays are shown in Fig. 9(a)–(c). SPICE simulations were performed to obtain SSN and worst case propagation delays for three different cases. In each of these test cases, 16 buffers were switching. In Case A, all the output buffers were switching simultaneously. In Case B, eight buffers are configured with the circuit shown in Fig. 9(a) and eight buffers are configured as tapered buffers with unit delay as shown in Fig. 9(b). In Case C, four simultaneously switching buffers each were configured to have zero to three delays. The SPICE simulation results of the switching noise and worst-case propagation delay are shown in Table I. The simulations are performed for four-stage and six-stage buffers designed for 10 pF capacitive load and a six-stage buffer designed for 100 pF capacitive load. The parasitic inductance was 1 nH for

all the simulations. It can be clearly seen that the switching noise reduces with increased skewness in buffer switching. As expected the worst-case propagation delay increased with skew. The variation in overall propagation delay with SSN can be observed from the comparison of four- and six-stage buffers driving the 10 pF capacitive load. The propagation delay of the four-stage buffer is smaller than the six-stage buffer for large number of simultaneously switching buffers as seen in Cases A and B. However, for Case C, the propagation delay of the six-stage buffer is smaller. This is due to greater alleviation of SSN effects in the bigger final inverter of the six-stage buffer due to skewing of output switching.

Additionally, the effects of overdamped/underdamped oscillations that occur in the RLC equivalent circuits can also be taken into consideration in skewing of output driver switching [34]. For low-power, low-voltage applications, the effects of SSN are expected to increase. This is because of increased current driving capacity (k_{eff}) that is required out of the transistors to satisfy the speed requirements with reduced supply voltages [35]. Therefore, it is necessary to extend the study of SSN of output buffers to internal circuitry of the integrated circuits.

VI. CONCLUSIONS AND FUTURE WORK

Studies on simultaneous switching noise are essential for reliable and optimal operation of tapered buffers. A new formulation for estimating SSN during the switching of inverters is derived. This formula includes the velocity-saturation effects seen in short-channel transistors. SSN reduces the number of tapered inverter stages and increases the taper factor for minimum overall buffer propagation delay. There is an optimum taper factor for buffer output transition time in the presence of SSN. SSN in tapered buffers can be reduced by skewing the switching of output buffers. However, this results in an increase in the worst case buffer propagation delay.

An optimization methodology need to be developed to improve the buffer performance in the presence of switching noise. Buffer optimization is important since they are one of the principal sources of current consumption and power dissipation in integrated circuits. Since, the magnitude of SSN reduces with smaller driving strength of the final inverter stages, variable-taper buffer design methodology that yields similar transistor sizes will be compared as an alternative to fixed-taper designs (output transition times in variable taper buffer inverter stages increases as one moves from the input inverter to the output inverter in tapered buffer chain) [13].

In addition to output switching, the contributions of internal circuit switching can also increase the noise. Although it is difficult to quantify, additional sources of noise such as substrate coupling and other local effects near the output buffers can contribute significant amount of noise.

ACKNOWLEDGMENT

The author would like to acknowledge the comments of the anonymous reviewers which resulted in an improved manuscript.

REFERENCES

- [1] H. C. Lin and L. W. Linholm, "An optimized output stage for MOS integrated circuits," *IEEE J. Solid-State Circuits*, vol. SC-10, pp. 106–109, Apr. 1975.
- [2] R. C. Jaeger, "Comments on 'An optimized output stage for MOS integrated circuits'," *IEEE J. Solid-State Circuits*, vol. SC-10, pp. 185–186, June 1975.
- [3] A. Kanuma, "CMOS circuit optimization," *Solid-State Electron.*, vol. 26, pp. 47–58, Jan. 1983.
- [4] N. Hedenstierna and K. Jeppson, "CMOS circuit speed and buffer optimization," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 270–281, Mar. 1987.
- [5] F. S. Lai, "A generalized algorithm for CMOS circuit delay, power, and area optimization," *Solid-State Electron.*, vol. 31, pp. 1619–1627, Nov. 1988.
- [6] N. C. Li, G. L. Haviland, and A. A. Tuszynski, "CMOS tapered buffer," *IEEE J. Solid-State Circuits*, vol. 25, pp. 1005–1008, Aug. 1990.
- [7] C. Prunty and L. Gal, "Optimum tapered buffer," *IEEE J. Solid-State Circuits*, vol. 27, pp. 118–119, Jan. 1992.
- [8] B. S. Cherkauer and E. G. Friedman, "Design of tapered buffers with local interconnect capacitance," *IEEE J. Solid-State Circuits*, vol. 30, pp. 151–155, Feb. 1995.
- [9] S. R. Vemuru, "Split inherent-capacitive load buffer design," *Int. J. Electron.*, vol. 79, pp. 359–365, Feb. 1995.
- [10] H. J. M. Veendrick, "Short-circuit power dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *IEEE J. Solid-State Circuits*, vol. SC-19, pp. 468–473, Aug. 1984.
- [11] L. A. Glasser and L. P. J. Hoyte, "Delay and power optimization in VLSI circuits," in *Proc. ACM/IEEE Design Automation Conf.*, pp. 529–535, 1984.
- [12] B. Hoppe, G. Nucundorf, D. Schmitt-Landseidel, and W. Specks, "Optimization of high-speed CMOS logic circuits with analytical models for signal delay, chip-area, and dynamic power dissipation," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 236–247, Mar. 1990.
- [13] S. R. Vemuru and A. R. Thorbjornsen, "Variable-taper CMOS buffer," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1265–1269, Sept. 1991.
- [14] M. Sha and P. Franzon, "Energy control and accurate delay estimation in the design of CMOS buffers," *IEEE J. Solid-State Circuits*, vol. 29, pp. 1150–1153, Sept. 1994.
- [15] J. S. Choi and K. Lee, "Design of CMOS tapered buffers with minimum power-delay product," *IEEE J. Solid-State Circuits*, vol. 29, pp. 1142–46, Sept. 1994.
- [16] B. S. Cherkauer and E. G. Friedman, "A unified design methodology for CMOS tapered buffers," *IEEE Trans. VLSI Syst.*, vol. 3, pp. 99–110, Mar. 1995.
- [17] C. M. Lee and H. Soukup, "An algorithm for CMOS timing and area optimization," *IEEE J. Solid-State Circuits*, vol. SC-19, pp. 781–787, Oct. 1984.
- [18] E. T. Lewis, "Optimization of device delay and overall area for CMOS VLSI designs," *Proc. IEEE*, vol. 72, pp. 670–688, June 1984.
- [19] A. J. Al-Khalili, Y. Zhu, and D. Al-Khalili, "A module generator for optimized CMOS buffers," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 1028–1046, Oct. 1990.
- [20] S. R. Vemuru and E. D. Smith, "Split-capacitive load variable taper buffer design," in *Proc. 34th Midwest Symp. Circuits Syst.*, 1991, pp. 815–818.
- [21] C. A. Neugebauer and R. O. Carlson, "Comparison of wafer scale integration with VLSI packaging approaches," *IEEE Trans. Comp., Hybrids, Manufact. Technol.*, vol. CHMT-10, pp. 184–189, June 1987.
- [22] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Trans. VLSI Syst.*, vol. 3, pp. 49–58, Mar. 1995.
- [23] S. R. Vemuru, "Simultaneous switching noise estimation for ASIC's," in *Proc. ASIC Conf.*, 1995, pp. 7–10.
- [24] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, pp. 584–594, Apr. 1990.
- [25] I. Shichman and D. A. Hodges, "Modeling and simulation of insulated-gate field-effect transistor switching circuits," *IEEE J. Solid-State Circuits*, vol. SC-3, pp. 285–289, Sept. 1968.
- [26] S. R. Vemuru and N. Scheinberg, "Short-circuit dissipation estimation for CMOS logic gates," *IEEE Trans. Circuits Syst. I*, vol. 41, pp. 762–765, Nov. 1994.
- [27] Y. P. Tsividis, *Operation and the Modeling of the MOS Transistor*. New York: McGraw-Hill, 1987.
- [28] R. Senthinathan and J. L. Prince, "Simultaneous switching ground noise calculation for packaged CMOS devices," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1724–1728, Nov. 1991.

- [29] A. Vaidyanath, B. Thoroddsen, and J. L. Prince, "Effect of CMOS driver loading conditions on simultaneous switching noise," *IEEE Trans. Comp., Packag., Manufact. Technol.*, vol. 17, pp. 480-485, Nov. 1994.
- [30] S. R. Vemuru, "Accurate simultaneous switching noise estimation including velocity-saturation effects," *IEEE Trans. Comp., Packag., Manufact. Technol. B*, vol. 19, pp. 344-349, May 1996.
- [31] M. Shoji, *Theory of CMOS Digital Circuits and Circuit Failures*. Princeton, NJ: Princeton Univ. Press, 1992.
- [32] K. D. Wagner and E. J. McCluskey, "Effect of supply voltage on circuit propagation delay and test applications," in *Tech. Dig. Int. Conf. Computer-Aided Design*, pp. 42-44, 1985.
- [33] L. Benini, M. Favalli, and B. Ricco, "Analysis of hazard contribution to power dissipation in CMOS IC's," in *IEEE Int. Workshop on Low Power Design*, 1994, pp. 327-332.
- [34] R. Senthinathan, J. L. Prince, and S. Nimmagadda, "Noise Immunity Characteristics of CMOS receivers and effects of skewing/damping CMOS output driver switching waveform on the simultaneous switching noise," *Microelectron. J.*, vol. 23, pp. 29-36, Mar. 1992.
- [35] D. Singh, J. Rabaez, M. Pedram, F. Catthoor, S. Rajgopal, N. Schgal, and T. Mozdzen, "Power conscious CAD tools and methodologies: A perspective," *Proc. IEEE*, vol. 83, pp. 570-594, Apr. 1995.



Srinivasa R. Vemuru (S'87-M'91) was born in Nellore, India, on December 15, 1962. He received the B.S. and the M.S. degrees in electrical engineering from the Indian Institute of Technology, Madras, India, and the Ph.D. degree in engineering science from the University of Toledo, Toledo, OH, in 1984, 1986, and 1991, respectively.

Since 1991, he has been an Assistant Professor in the Electrical Engineering Department of the City College and the Graduate Center of the City University of New York. His research interests are

in the areas of analog and digital circuit design and test and CAD tools for VLSI.

Dr. Vemuru has served on the program committees of the International Conference on Electronics, Circuits, and Systems and the International Conference on Modeling and Simulation. He is a member of IEEE-CS, IEEE-CAS, ACM-SIGDA, Phi Kappa Phi, and Sigma Xi.